

# Macro Help Contents

WordPerfect's Macro Reference Help provides information on the items listed here. The command explanations are under the alphabetic listing. To access these items, click on the appropriate title or letter below.

[A](#)   [B](#)   [C](#)   [D](#)   [E](#)   [F](#)   [G](#)   [H](#)   [I-K](#)   [L](#)   [M](#)   [N-O](#)   [P](#)   [Q-R](#)   [S](#)   [T](#)  
[U-Z](#)

[How to Use the Macro Help System](#)

[Macro Basics](#)

[Programming and Product Commands](#)

[What are Loops?](#)

[How do you create Dialog Boxes?](#)

[Macro Substitution](#)

[List of Shipping Macros](#)

# How to Use the Macro Help System

This help system is designed to make learning the WordPerfect for Windows Macro commands easy. The information in this file is divided alphabetically. Programming commands are listed first, followed by Product commands.

When you click on a command name, an explanation of the command is displayed, followed by the command parameters and the command syntax.

Examples are provided for many commands. Each example shows how a command can be used in a macro. To access an example, click on the words Show Me! at the bottom of the window. You can copy an example by choosing Copy from the Edit menu in the help window. You can then paste the example into your own macro by choosing Paste from the Edit menu in the active document window.

Pop-up glossary definitions appear when you click on words or phrases marked in green with a dotted underline. To make the definition disappear, click the mouse again anywhere.

Related references are listed in green with an underline at the bottom of many help windows. To view the related command, simply click on it.

In addition, this file allows you to access all of the Windows help menu items, including the Help Window Buttons (Contents, Search, Back, History, and Browse).

## Macro Help Index: A

[A](#)   [B](#)   [C](#)   [D](#)   [E](#)   [F](#)   [G](#)   [H](#)   [I-K](#)   [L](#)   [M](#)   [N-O](#)   [P](#)   [Q-R](#)   [S](#)   [T](#)   [U-Z](#)

[APPLICATION](#)  
[ASSERTCANCEL](#)  
[ASSERTERROR](#)  
[ASSERTNOTFOUND](#)  
[ASSIGN](#)  
[AboutDlg](#)  
[Advance](#)  
[AdvanceDlg](#)  
[AppMaximize](#)  
[AppMinimize](#)  
[AppMove](#)  
[AppRestore](#)  
[AppSize](#)

## Macro Help Index: B

A   B   C   D   E   F   G   H   I-K   L   M   N-O   P   Q-R   S   T   U-Z

BEEP  
BYTELEN  
BYTEPOS  
BlockProtect  
BoxEditCaption  
BoxEditor  
BoxFigureEdit  
BoxNewNumber  
BoxOptions  
BoxPosition  
BoxRetrieve  
BoxSelect  
ButtonBarAddItem  
ButtonBarDeleteItem  
ButtonBarEditDlg  
ButtonBarNewDlg  
ButtonBarOptions  
ButtonBarOptionsDlg  
ButtonBarSave  
ButtonBarSaveDlg  
ButtonBarSelect  
ButtonBarSelectDlg  
ButtonBarShow

## Macro Help Index: C

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[//Comment](#)  
[CALL](#)  
[CANCELOFF](#)  
[CANCELON](#)  
[CASE](#)  
[CASECALL](#)  
[CHAIN](#)  
[CaptionBoxNumber](#)  
[CaseToLower](#)  
[CaseToUpper](#)  
[Close](#)  
[CloseNoSave](#)  
[ColumnDefine](#)  
[ColumnDefineDlg](#)  
[ColumnDefineEven](#)  
[ColumnsOff](#)  
[ColumnsOn](#)  
[CommentConvertToText](#)  
[CommentCreate](#)  
[CommentCreateDlg](#)  
[CommentEditDlg](#)  
[CommentShow](#)  
[ConditionalEOP](#)  
[ConditionalEOPDlg](#)  
[ControlId](#)  
[ConvertClipboardDlg](#)  
[ConvertClipboardPicture](#)  
[CrossRefMark](#)  
[CrossRefMarkDlg](#)

## Macro Help Index: D

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[DEFAULTUNITS](#)  
[DLLCALL](#)  
[DLLFREE](#)  
[DLLLOAD](#)  
[DIALOGBOXES](#)  
[DateCode](#)  
[DateFormat](#)  
[DateFormatDlg](#)  
[DateText](#)  
[DDECreateLink](#)  
[DDECreateLinkDlg](#)  
[DDEDeleteLink](#)  
[DDEDeleteLinkDlg](#)  
[DDEEditLink](#)  
[DDEEditLinkDlg](#)  
[DDEPasteLink](#)  
[DDEUpdateLink](#)  
[DDEUpdateLinkDlg](#)  
[DeleteCharNext](#)  
[DeleteCharPrevious](#)  
[DeleteEOL](#)  
[DeleteEOP](#)  
[DeleteToBeginningOfWord](#)  
[DeleteToEndOfWord](#)  
[DeleteWord](#)  
[DialogAddCheckBox](#)  
[DialogAddColorWheel](#)  
[DialogAddComboBox](#)  
[DialogAddCounter](#)  
[DialogAddEditBox](#)  
[DialogAddFileNameBox](#)  
[DialogAddFrame](#)  
[DialogAddGroupBox](#)  
[DialogAddHLine](#)  
[DialogAddHotSpot](#)  
[DialogAddIcon](#)  
[DialogAddListBox](#)  
[DialogAddListItem](#)  
[DialogAddPopUpButton](#)  
[DialogAddPushButton](#)  
[DialogAddRadioButton](#)  
[DialogAddScrollBar](#)  
[DialogAddText](#)  
[DialogAddViewer](#)  
[DialogAddVLine](#)  
[DialogDefine](#)  
[DialogDestroy](#)  
[DialogDisplay](#)  
[Display](#)

DisplayFontAdjustment  
DisplayPitch  
DisplayPitchDlg  
DocCompare  
DocCompareAddMarkingsDlg  
DocCompareRemove  
DocCompareRemoveMarkingsDlg  
DocInitialCodes  
DocInitialFont  
DocInitialFontDlg  
DocMaximize  
DocMinimize  
DocMove  
DocNext  
DocPrevious  
DocRedlineMethod  
DocRedlineMethodDlg  
DocRestore  
DocSize  
DocSummaryDefine  
DocSummaryDelete  
DocSummaryDlg  
DocSummaryExtract  
DocSummaryGetData  
DocSummaryPrint  
DocSummarySaveAs  
DraftMode

## Macro Help Index: E

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[ELSE](#)  
[ENDAPP](#)  
[ENDFOR](#)  
[ENDIF](#)  
[ENDPROMPT](#)  
[ENDWHILE](#)  
[ERROROFF](#)  
[ERRORON](#)  
[EditAppend](#)  
[EditCopy](#)  
[EditCut](#)  
[EditPaste](#)  
[EndnoteCreate](#)  
[EndnoteEdit](#)  
[EndnoteEditDlg](#)  
[EndnoteNewNumber](#)  
[EndnoteNewNumberDlg](#)  
[EndnoteOptions](#)  
[EndnoteOptionsDlg](#)  
[EndnotePlacement](#)  
[EndnotePlacementDlg](#)  
[EquationCaptionEdit](#)  
[EquationCreate](#)  
[EquationCreateDlg](#)  
[EquationEdit](#)  
[EquationNewNumberDlg](#)  
[EquationOptionsDlg](#)  
[EquationPalette](#)  
[EquationPositionDlg](#)  
[EquationSettings](#)  
[EquationSettingsDlg](#)  
[EquationZoom100](#)  
[EquationZoom200](#)  
[EquationZoomFill](#)  
[EquationZoomIn](#)  
[EquationZoomOut](#)  
[ExecuteFunction](#)



## Macro Help Index: F

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[FOR](#)  
[FOREACH](#)  
[FRACTION](#)  
[FigureCaptionEdit](#)  
[FigureCreate](#)  
[FigureCreateDlg](#)  
[FigureEdit](#)  
[FigureNewNumberDlg](#)  
[FigureOptionsDlg](#)  
[FigurePositionDlg](#)  
[FigureRetrieveDlg](#)  
[FileChangeDir](#)  
[FileCopy](#)  
[FileDelete](#)  
[FileFindDlg](#)  
[FileManager](#)  
[FileMove](#)  
[FileNew](#)  
[FileOpen](#)  
[FileOpenDlg](#)  
[FileRetrieve](#)  
[FileRetrieveDlg](#)  
[FileSave](#)  
[FileSaveAsDlg](#)  
[Font](#)  
[FontBold](#)  
[FontColor](#)  
[FontColorDlg](#)  
[FontDlg](#)  
[FontDoubleUnderline](#)  
[FontExtraLarge](#)  
[FontFine](#)  
[FontItalic](#)  
[FontLarge](#)  
[FontNormal](#)  
[FontOutline](#)  
[FontRedline](#)  
[FontShadow](#)  
[FontSmall](#)  
[FontSmallCaps](#)  
[FontStrikeout](#)  
[FontSubscript](#)  
[FontSuperscript](#)  
[FontUnderline](#)  
[FontVeryLarge](#)  
[FooterDlg](#)  
[FootnoteCreate](#)  
[FootnoteEdit](#)  
[FootnoteEditDlg](#)

FootnoteNewNumber  
FootnoteNewNumberDlg  
FootnoteOptions  
FootnoteOptionsDlg  
ForcePageEven  
ForcePageOdd

## Macro Help Index: G

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[GETNUMBER](#)

[GETSTRING](#)

[GETUNITS](#)

[GO](#)

[Generate](#)

[GenerateDlg](#)

[GetWPData](#)

[Grammatik](#)

[GraphicLine](#)

[GraphicLineSelect](#)

[GraphicsShow](#)

## Macro Help Index: H

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[HardHyphen](#)  
[HardPageBreak](#)  
[HardPageBreakInsert](#)  
[HardReturn](#)  
[HardReturnInsert](#)  
[HardSpace](#)  
[HardTabCenter](#)  
[HardTabCenterDot](#)  
[HardTabDecimal](#)  
[HardTabDecimalDot](#)  
[HardTabDecimalInsert](#)  
[HardTabLeft](#)  
[HardTabLeftDot](#)  
[HardTabRight](#)  
[HardTabRightDot](#)  
[HeaderDlg](#)  
[HeaderFooter](#)  
[HeaderFooterPlacement](#)  
[HeaderFooterPlacementDlg](#)  
[HelpContextSensitive](#)  
[HelpGlossary](#)  
[HelpHowDoI](#)  
[HelpIndex](#)  
[HelpKeyboard](#)  
[HelpUsingMacros](#)  
[HelpUsingWPHelp](#)  
[HelpWhatIs](#)  
[HorizLineCreateDlg](#)  
[HorizLineEditDlg](#)  
[HorizScrollBarShow](#)  
[Hyphen](#)  
[HyphenIgnoreWord](#)  
[HyphenSoft](#)  
[HyphenSoftReturn](#)

## Macro Help Index: I-K

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[I](#)

[INTEGER](#)

[IndexDefine](#)

[IndexDefinedDlg](#)

[IndexMark](#)

[IndexMarkDlg](#)

[InhibitInput](#)

[InsertSpecialCodesDlg](#)

[InsertTypeover](#)

[JustifyCenter](#)

[JustifyFull](#)

[JustifyLeft](#)

[JustifyRight](#)

[KerningSpacing](#)

[KeyboardSelect](#)

[KeyboardSelectDlg](#)

## Macro Help Index: L

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[LABEL](#)

[Language](#)

[LanguageDlg](#)

[LastOpenedAdd](#)

[LineCenter](#)

[LineCenterEnd](#)

[LineDraw](#)

[LineDrawDlg](#)

[LineFlushRight](#)

[LineHeight](#)

[LineHeightDlg](#)

[LineHyphenation](#)

[LineHyphenationDlg](#)

[LineNumbering](#)

[LineNumberingDlg](#)

[LineSpacing](#)

[LineSpacingDlg](#)

[ListDefine](#)

[ListDefineDlg](#)

[ListMark](#)

[ListMarkDlg](#)

## Macro Help Index: M

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[MENU](#)

[MacroAssignDlg](#)

[MacroDialogResult](#)

[MacroMenuAppend](#)

[MacroMenuDelete](#)

[MacroStatusPrompt](#)

[MailMAPIDlg](#)

[MailVIMDlg](#)

[MailWPDlg](#)

[MarginRelease](#)

[MarginReleaseInsert](#)

[MasterDocCondense](#)

[MasterDocCondenseDlg](#)

[MasterDocExpand](#)

[MasterDocSubdocDlg](#)

[MasterDocSubdocInsert](#)

[MenuAddItem](#)

[MenuDeleteItem](#)

[MergeCodesDlg](#)

[MergeConvert](#)

[MergeConvertMsgDlg](#)

[MergeDOSText](#)

[MergeEndField](#)

[MergeEndRecord](#)

[MergeExecute](#)

[MergeFieldDlg](#)

[MergeFilesDlg](#)

[MergeInputDlg](#)

[MergeInsertCode](#)

[MergeNextRecord](#)

[MergePageOff](#)

[MergeVariableGet](#)

[MergeVariableSet](#)

## Macro Help Index: N-O

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[NEWDEFAULT](#)  
[NUMSTR](#)  
[NoteNext](#)  
[NoteNumber](#)  
[NotePrevious](#)  
[NumberFormat](#)

[ONCANCEL](#)  
[ONCANCEL CALL](#)  
[ONERROR](#)  
[ONERROR CALL](#)  
[ONNOTFOUND](#)  
[ONNOTFOUND CALL](#)  
[OleInsertObject](#)  
[OleInsertObjectDlg](#)  
[OleLinksDlg](#)  
[OutlineCopy](#)  
[OutlineDefine](#)  
[OutlineDelete](#)  
[OutlineMove](#)  
[OutlineOff](#)  
[OutlineOn](#)  
[OverstrikeCreate](#)  
[OverstrikeCreateDlg](#)  
[OverstrikeEdit](#)  
[OverstrikeEditDlg](#)



## Macro Help Index: P

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[PAUSE](#)  
[PROMPT](#)  
[PageCenter](#)  
[PageMargins](#)  
[PageMarginsDlg](#)  
[PageNumbering](#)  
[PageNumberingDlg](#)  
[PageNumberInsert](#)  
[PageSuppress](#)  
[PageSuppressDlg](#)  
[PageWidowOrphan](#)  
[PaperSizeAdd](#)  
[PaperSizeDelete](#)  
[PaperSizeDlg](#)  
[PaperSizeListCreate](#)  
[PaperSizeListFree](#)  
[PaperSizeListGetItem](#)  
[PaperSizeSelect](#)  
[ParagraphDoubleIndent](#)  
[ParagraphHangingIndent](#)  
[ParagraphIndent](#)  
[ParagraphNumber](#)  
[ParagraphNumberDefDlg](#)  
[ParagraphNumberingDlg](#)  
[PauseKey](#)  
[PosCellDown](#)  
[PosCellNext](#)  
[PosCellPrevious](#)  
[PosCellUp](#)  
[PosCharNext](#)  
[PosCharPrevious](#)  
[PosColumnNext](#)  
[PosColumnPrevious](#)  
[PosDocBottom](#)  
[PosDocTop](#)  
[PosDocVeryTop](#)  
[PosGoTo](#)  
[PosGoToDlg](#)  
[PosLineBegin](#)  
[PosLineDown](#)  
[PosLineEnd](#)  
[PosLineUp](#)  
[PosLineVeryBegin](#)  
[PosLineVeryEnd](#)  
[PosPageBottom](#)  
[PosPageNext](#)  
[PosPagePrevious](#)  
[PosPageTop](#)  
[PosParagraphNext](#)

PosParagraphPrevious  
PosScreenDown  
PosScreenLeft  
PosScreenRight  
PosScreenUp  
PosTableBegin  
PosTableColumnBottom  
PosTableColumnTop  
PosTableEnd  
PosTableRowBegin  
PosTableRowEnd  
PosWordNext  
PosWordPrevious  
PrefBackup  
PrefBackupDlg  
PrefBeep  
PrefDateFormat  
PrefDateFormatDlg  
PrefDisplayDlg  
PrefDisplaySet  
PrefDocSummary  
PrefDocSummaryDlg  
PrefDraftColors  
PrefDraftColorsDlg  
PrefEnvSettings  
PrefEnvSettingsDlg  
PrefEquation  
PrefEquationDlg  
PrefHyphenation  
PrefInitialCodes  
PrefLocationDlg  
PrefLocationOfFiles  
PrefMenu  
PrefMergeDelimit  
PrefMergeDlg  
PrefPrintCopies  
PrefPrintDoc  
PrefPrintRedline  
PrefPrintSettingsDlg  
PrefPrintSizeRatio  
PrefPrintWindowDriverSet  
PrefRevealCodesColors  
PrefRevealCodesColorsDlg  
PrefRuler  
PrefSave  
PrefToA  
PrefToADlg  
PrefZoom  
PrefZoomDlg  
PrintDlg  
PrintDoc  
PrinterCommand  
PrinterCommandDlg  
PrinterInitialize  
PrinterSelect

PrinterSelectDlg  
PrinterSelectListCreate  
PrinterSelectListFree  
PrinterSelectListGetItem  
PrintFull  
PrintMultiplePages  
PrintOptions  
PrintPage  
PrintPreview  
PrintSelected

## Macro Help Index: Q-R

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[QUIT](#)

[REPEAT](#)

[RETURN](#)

[RETURNCANCEL](#)

[RETURNERROR](#)

[RETURNNOTFOUND](#)

[RUN](#)

[Redisplay](#)

[RevealCodes](#)

[RulerShow](#)

## Macro Help Index: S

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[SPEED](#)  
[STRLEN](#)  
[STRNUM](#)  
[STRPOS](#)  
[SUBBYTE](#)  
[SUBSTR](#)  
[ScrollScreenLeft](#)  
[ScrollScreenRight](#)  
[SearchDlg](#)  
[SearchNext](#)  
[SearchPrevious](#)  
[SearchReplace](#)  
[SearchReplaceDlg](#)  
[SearchText](#)  
[SelectAll](#)  
[SelectCell](#)  
[SelectCellDown](#)  
[SelectCellDownArrow](#)  
[SelectCellLeft](#)  
[SelectCellRight](#)  
[SelectCellUp](#)  
[SelectCellUpArrow](#)  
[SelectCharNext](#)  
[SelectCharPrevious](#)  
[SelectColumn](#)  
[SelectColumnBottom](#)  
[SelectColumnNext](#)  
[SelectColumnPrevious](#)  
[SelectColumnTop](#)  
[SelectDocBottom](#)  
[SelectDocTop](#)  
[SelectDocVeryTop](#)  
[SelectLineBegin](#)  
[SelectLineDown](#)  
[SelectLineEnd](#)  
[SelectLineUp](#)  
[SelectLineVeryBegin](#)  
[SelectLineVeryEnd](#)  
[SelectMode](#)  
[SelectPage](#)  
[SelectPageNext](#)  
[SelectPagePrevious](#)  
[SelectParagraph](#)  
[SelectParagraphNext](#)  
[SelectParagraphPrevious](#)  
[SelectRectangle](#)  
[SelectScreenDown](#)  
[SelectScreenLeft](#)  
[SelectScreenRight](#)

SelectScreenUp  
SelectSentence  
SelectSentenceNext  
SelectSentencePrevious  
SelectTable  
SelectTableColumn  
SelectTableColumnExtendLeft  
SelectTableColumnExtendRight  
SelectTableRow  
SelectWord  
SelectWordNext  
SelectWordPrevious  
ShortMenus  
Sort  
SortDlg  
Speller  
SpreadsheetImportDlg  
SpreadsheetImportLink  
SpreadsheetLinkCreateDlg  
SpreadsheetLinkEditDlg  
SpreadsheetLinkOptions  
SpreadsheetLinkOptionsDlg  
SpreadsheetLinkUpdateAll  
SpreadsheetLinkUpdateAllDlg  
Styles  
StylesDelete  
StylesDlg  
StylesEdit  
StylesProperties  
StylesPropertiesDlg  
StylesRetrieve  
StylesSave

## Macro Help Index: T

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[Tab](#)  
[TabInsert](#)  
[TableAppendRow](#)  
[TableBoxCaptionEdit](#)  
[TableBoxCreate](#)  
[TableBoxCreateDlg](#)  
[TableBoxEdit](#)  
[TableBoxNewNumberDlg](#)  
[TableBoxOptionsDlg](#)  
[TableBoxPositionDlg](#)  
[TableCalculate](#)  
[TableCell](#)  
[TableCellDlg](#)  
[TableColumn](#)  
[TableColumnDlg](#)  
[TableConvert](#)  
[TableConvertDlg](#)  
[TableCreate](#)  
[TableCreateNewDlg](#)  
[TableDeleteColumn](#)  
[TableDeleteDlg](#)  
[TableDeleteRow](#)  
[TableEditColumn](#)  
[TableFormula](#)  
[TableFormulaDlg](#)  
[TableInsertColumn](#)  
[TableInsertDlg](#)  
[TableInsertRow](#)  
[TableJoin](#)  
[TableLineDlg](#)  
[TableLines](#)  
[TableOptions](#)  
[TableOptionsDlg](#)  
[TableRow](#)  
[TableRowDlg](#)  
[TableSplit](#)  
[TableSplitDlg](#)  
[TabSet](#)  
[TabSetDlg](#)  
[TextBoxCaptionEdit](#)  
[TextBoxCreate](#)  
[TextBoxCreateDlg](#)  
[TextBoxEdit](#)  
[TextBoxNewNumberDlg](#)  
[TextBoxOptionsDlg](#)  
[TextBoxPositionDlg](#)  
[TextBoxRotate](#)  
[TextBoxRotateDlg](#)  
[Thesaurus](#)

ToADefine  
ToADefineDlg  
ToAEditFull  
ToAMarkFull  
ToAMarkShort  
ToAMarkShortDlg  
ToCDefine  
ToCDefineDlg  
ToCMark  
ToCMarkDlg  
Type  
TypeChar  
TypesetBaseline  
TypesetDlg  
TypesetJustifyLimits  
TypesetKerning  
TypesetLeadingAdjust  
TypesetLetterspace  
TypesetManualKerningDlg  
TypesetUnderlineOptions  
TypesetWordSpace



## Macro Help Index: U-Z

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I-K](#) [L](#) [M](#) [N-O](#) [P](#) [Q-R](#) [S](#) [T](#) [U-Z](#)

[UNTIL](#)

[Undelete](#)

[UndeleteDlg](#)

[Undo](#)

[UserBoxCaptionEdit](#)

[UserBoxCreate](#)

[UserBoxCreateDlg](#)

[UserBoxEdit](#)

[UserBoxNewNumberDlg](#)

[UserBoxOptionsDlg](#)

[UserBoxPositionDlg](#)

[User-defined Dialog Boxes](#)

[UserFunction](#)

[VertLineCreateDlg](#)

[VertLineEditDlg](#)

[WAIT](#)

[WHILE](#)

[WindowCascade](#)

[WindowTile](#)

[WordCountDlg](#)

[WPCharactersDlg](#)

[Zoom](#)

[Zoom100](#)

[Zoom150](#)

[Zoom200](#)

[Zoom50](#)

[Zoom75](#)

[ZoomDlg](#)

[ZoomToPageWidth](#)

---

## Macro Basics

A macro is a single command or a set of commands that executes a series of instructions. Macros are often used to perform repetitive tasks quickly, such as creating a mailing address or a closing for a letter.

This lesson will tell you how to do the following:

[Record a Macro](#)

[Play a Macro](#)

[Edit a Macro](#)

[Assign a Macro to the Macro Menu](#)

[Assign a Macro to the Button Bar](#)

[Assign a Macro to a Keystroke](#)

[Errors in Macros](#)

---

## Record a Macro

To record a macro,

- 1 Choose Record from the Macro menu.
- 2 Type a filename for the macro in the Filename text box. If you do not specify a filename extension, WordPerfect attaches .wcm extension to the filename of any macro you create.
- 3 Type a description of the macro in the Descriptive Name text box.
- 4 Choose Record to return to the document window and start recording the macro.

The "Recording Macro" message appears on the status bar whenever WordPerfect is recording a macro.

You cannot use the mouse to position the insertion point while recording a macro. However, you can use the mouse to do anything else in the program when you record.

- 5 Choose Stop from the Macro menu to stop recording the macro.

If you did a basic installation, WordPerfect created a directory for your macros (C:\WPWIN\MACROS). The macro is now saved in this directory. If you do not have a macros directory specified, the macro is saved in the current document directory. If no macro directory or document directory is specified, the macro is saved in the same directory as WPWIN.EXE (the main WordPerfect program file).

[Play a Macro](#)

[Edit a Macro](#)

[Assign a Macro to the Macro Menu](#)

[Assign a Macro to the Button Bar](#)

[Assign a Macro to a Keystroke](#)

[Errors in Macros](#)

---

## Play a Macro

To play a macro:

- 1 Choose Play from the Macro menu.
- 2 Find the filename of the macro you want to play.
- 3 Double-click the macro in the Macros directory, or select it and then choose Play.

The macro plays, beginning at the insertion point.

[Record a Macro](#)

[Edit a Macro](#)

[Assign a Macro to the Macro Menu](#)

[Assign a Macro to the Button Bar](#)

[Assign a Macro to a Keystroke](#)

[Errors in Macros](#)

---

## **Edit a Macro**

You may want to change a macro from time to time for various reasons.

WordPerfect for Windows macros are regular document files (with a compiled macro object hidden at the beginning of the file). This means that you can write, edit, and save macros just the same as you would any other WordPerfect document.

To edit a macro,

- 1 Open the file containing your macro into a WordPerfect document window.

The text of the macro appears as a series of commands. A command is a word or series of words that instructs WordPerfect to perform a certain action.

- 2 Add or delete commands and text from your macro.

WordPerfect macros use a "free form" editor. When you record or edit a macro, you can format the macro any way you like.

When you finish editing your macro, you can close and save it.

- 3 Choose Close from the File menu, then choose Yes to save the changes you made to the macro.

[Record a Macro](#)

[Play a Macro](#)

[Assign a Macro to the Macro Menu](#)

[Assign a Macro to the Button Bar](#)

[Assign a Macro to a Keystroke](#)

[Errors in Macros](#)

---

## Errors in Macros

The first time you play a macro after you record or edit it, WordPerfect "compiles" the macro. This means that the macro text is condensed into code that the computer can read quickly.

If you make a mistake when recording or editing the macro, compilation stops and the WordPerfect Macro Facility - Syntax Error dialog box appears when you try to play the macro.

This dialog box can help you locate your mistakes. The Description of Problem box shows you the error in your macro. The Location of Error box shows you where the problem occurred. Once you've made a note of the problem, you can choose Continue Compilation if you want to look for more errors, or Cancel Compilation if you want to go back and fix the problem you just found. In either case, the macro will not play after the dialog box closes. You must fix the errors in the macro before it will play.

Although the compiler will give you a good idea of where the error is, you may need to search around a bit to find exactly where you need to make a change in your macro. The macro language is complex, so the compiler sometimes has to make a "guess" at what you were trying to do. If you search the area the WordPerfect Macro Facility - Syntax Error dialog box pinpoints, you should be able to find the error. Pay special attention to missing parentheses and semicolons; these make up the majority of syntax errors.

[Record a Macro](#)

[Play a Macro](#)

[Edit a Macro](#)

[Assign a Macro to the Macro Menu](#)

[Assign a Macro to the Button Bar](#)

[Assign a Macro to a Keystroke](#)

---

## Assign a Macro to the Macro Menu

You can assign macros to the Macro menu to make them easier to play. To play one of these macros, you simply choose it from the Macro menu.

To assign a macro to the Macro menu,

- 1 Choose Assign to Menu from the Macro menu.

The Assign Macro to Menu dialog box appears. The Menu Text box lists the macros assigned to the Macro menu. As many as nine macros can be in the menu at a time.

- 2 Choose Insert to display the Insert Macro Menu Item dialog box.

In this dialog box, you specify the filename of the macro and how it will be described in the Macro menu.

- 3 Click the List button to display the list of files in the Macros directory.
- 4 Double-click or highlight the filename of your macro and then choose Select.

The macro's full pathname appears in the Macro Name text box.

- 5 Press Tab to move the insertion point to the Menu Text text box.

The descriptive name you typed when you recorded the macro is selected in the Menu Text text box.

- 6 Choose OK to return to the Assign Macro to Menu dialog box, then choose OK again to return to the document window.

Note that the name of the macro, not the macro itself, is saved in the Macro menu. If you edit your macro, the edited version will play when you choose the macro from the Macro Menu.

Record a Macro

Play a Macro

Edit a Macro

Assign a Macro to the Button Bar

Assign a Macro to a Keystroke

Errors in Macros

---

## Assign a Macro to the Button Bar

You can assign a macro to the Button Bar. This allows you to play the macro by simply clicking on the button.

To assign a macro to the Button Bar,

- 1 Choose Button Bar Setup from the View menu, then choose Edit.
- 2 Choose Assign Macro to Button.

The list of macros in the Macros directory appears.

- 3 Double-click on the filename of your macro or select it and then choose Assign.

A button for the macro appears at the end of the Button Bar.

- 4 Choose OK to return to the document window.

When Macro on Disk is selected, the macro's filename is saved to the Button Bar. If you then make changes to your macro, choosing the button will play the edited macro. If you clear the Macro on Disk checkbox, WordPerfect will not play subsequent versions of the macro.

[Record a Macro](#)

[Play a Macro](#)

[Edit a Macro](#)

[Assign a Macro to the Macro Menu](#)

[Assign a Macro to a Keystroke](#)

[Errors in Macros](#)



---

## Assign a Macro to a Keystroke

You can assign a macro to a keystroke on a "soft keyboard," so you can play the macro by simply pressing a key or a combination of keys.

Note to WordPerfect for DOS macro writers: WordPerfect for Windows does not have an Alt-key macro naming capability for alphanumerics; however, you can assign macros to Alt and Alt+Shift with function keys and all other combinations. You can also assign macros to a keystroke, Shift+keystroke, Ctrl+keystroke, and Ctrl+Shift+keystroke combinations for alphanumerics.

To assign a macro to a keystroke,

- 1 Choose Preferences from the File menu, then choose Keyboard.
- 2 Choose Create to display the Keyboard Editor dialog box.

If you have a different soft keyboard already defined, you can choose Edit to make the macro part of that soft keyboard.

- 3 Choose Macros from the Item Types pop-up list.

Now you're ready to specify what macros you want to assign to a keystroke.

- 4 Choose Add to display the Import Macro to Keyboard dialog box.
- 5 Double-click or select the filename of your macro and then choose Import.

The Import Macro to Keyboard dialog box disappears and the macro name appears selected in the Assignable Items list box. You can now assign a keystroke to the macro.

You can assign as many macros to the keyboard as you like.

- 6 Choose OK to close the Keyboard Editor dialog box.
- 7 If you are creating a soft keyboard, type a name for the keyboard in the Save As text box, then choose Save.

The Save Keyboard File dialog box is displayed. You'll need to save your keyboard remappings in a file.

- 8 Choose OK to make your remapped keyboard active.

Note that a copy of the macro is saved in the soft keyboard. This means that if you make changes to the macro, you will need to remove the macro from the keyboard, then reassign the macro to the key.

[Record a Macro](#)

[Play a Macro](#)

[Edit a Macro](#)

[Assign a Macro to the Macro Menu](#)

[Assign a Macro to the Button Bar](#)

[Errors in Macros](#)

---

## Programming and Product Commands

The WordPerfect macro language is divided into two parts: product commands and programming commands. These two types of commands let you use all of the functionality of WordPerfect.

In order to help you distinguish product commands from programming commands throughout the reference explanations, programming commands are written in all uppercase letters (BYTELEN) and product commands are written in a combination of upper and lowercase letters (MergeVariableGet). The macro facility, however, is not case sensitive; you may write macro commands in any combination of upper and lower case you choose. To emphasize that the macro facility is not case sensitive, macro commands in the examples are written in all lowercase letters.

The topics listed below provide explanations about programming and product commands:

[Understanding Syntax](#)

[What is a Product Command](#)

[What is a Programming Command](#)

---

## Understanding Syntax

Syntax means the spelling and structure of a command. A command must be spelled exactly for the macro command interpreter to understand it. If a command is spelled incorrectly, it will not be read by the WordPerfect macro interpreter and the macro will not work. Additionally, if the syntax or organization of the command is incomplete, the macro interpreter will not be able to run the macro.

The syntax of a command often includes certain elements called parameters. Some commands are simple enough that they need no clarification. Hard Return ( ) is one of these. Others, however, need parameters to specify exactly how the command is to function. Parameters may be used to turn a feature on or off or to specify how far from the edge of a page to set a margin or what size font to use.

In many product commands, a parameter specifies the structure of a command through value set members. Value set members are one of the most common and easy-to-use parameter types in macros. They consist of a member, which is the name of the function, and a numeric equivalent. When you use these parameters, you can specify the function using either the member or the numeric equivalent. For example, in the product command, PosGoTo, the command has parameters which allow you to position the insertion point on a certain page, within a document, or in a table. To specify where to position the insertion point, there is a **Where** parameter with value set members such as:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

TopofPage!	2
BottomofPage!	3

(In the help file, when value set members are used in a parameter, they are typically displayed in a table.)

When you use this command, you must be sure to include not only the command name (PosGoTo), but also the parameters and value set members. Your command might look something like this:

PosGoTo (Where:TopOfPage!;;)

or, if you use the numeric equivalent:

PosGoTo (Where:2;;)

---

## What is a Product Command?

A product command is a command that performs a function that an application would normally do. For example, the Type ( ) command types information and the HardReturn ( ) command inserts a Hard Return code ([HRt]) into a document

There are two ways you can find the product command name for a WordPerfect feature. The first is to click on Contents and browse through the alphabetic listing of commands until you find what you want. The second is to click on Search and type in the name of the command. Even if you do not know the full name of the command, when you type in part of the name, the Search function will display all commands listed in the index which are similar to what you typed. Then you simply double click on the desired topic or select the topic and click on Go To.

There are hundreds of product commands, one for almost every action you can perform in WordPerfect. We've taken care to make the names match their functions as closely as possible, but sometimes you may still have difficulties finding the right command for the function you want the macro to perform. If you can't locate a command, you can usually find it by recording, then opening, a test macro. With practice, you'll gain a knack for finding and remembering product commands.

The syntax for product commands often includes a named parameter. A named parameter is one that includes both the parameter name and the value for the parameter in the syntax of the command. The parameter name remains constant; to tailor the command for your macro, you change the value following the parameter name. In the Font command below **Font** is the command name, **Name** is a parameter name followed by the value which is a character expression. The parameter name and value are separated by a colon. The parameter name, **Name**, will be included in the syntax of the command every time the command is used. The value following it, the character expression, will change depending on the name of the font chosen. There are two other parameters, Size and ForcelInsert. These are also named parameters and are followed by a colon and a value. Unless otherwise directed in the command explanation, you will need to specify a value for each parameter in the command syntax.

### **Syntax:**

Font (Name:Character Expression; Size:Measurement Expression; ForcelInsert:Value Set Member)

### **Example:**

Font(Name:"CG Times";Size:12.0p;ForcelInsert:Yes!)

All product commands in this reference are typed in a combination of upper and lower case letters. This is simply to help you distinguish product commands from programming commands. You can actually type any programming or product commands in whatever combination of upper and lowercase letters you choose.

---

## What is a Programming Command?

Product commands allow you to perform any function that WordPerfect performs. In many cases, however, you may want to do more than simply set a font or insert a header into a document. You may wish to pause a macro to request information from the user, such as the name of the recipient of a letter. You may wish to perform a search-and-replace a specified number of times. Product commands are not capable of handling these tasks alone. Programming commands are needed to give you more control over what happens in the macro.

Product commands allow you to use WordPerfect features. Programming commands allow you to control how those features act and interact. Some of the most common uses for programming commands are to create macros that loop, macros that connect with other macros, macros that request and update information, macros that use dialog boxes, and macros that take advantage of Dynamic Data Exchange links.

All programming commands in the Macros Help Index screens are typed in uppercase letters. This is simply to help you distinguish product commands from programming commands. You can actually type any programming or product command in whatever combination of upper and lowercase letters you wish.

Loops

Connecting Macros

Dialog Boxes

Dynamic Data Exchange Links

---

## Loops

A "loop" is a term meaning a series of commands that repeats a specified number of times. Loops are created with programming commands and are useful in macros if you wish to perform a function repeatedly but you do not need to supervise every step of the process. For example, you may have your macro perform a search and delete on all occurrences of a word. Loops are also useful for gathering information. You can create a macro that brings up a dialog box designed to solicit mailing addresses which reappears until the user clicks on a "No more addresses" button.

A loop may repeat until a pre-set condition has been satisfied (REPEAT-UNTIL), while a condition exists (WHILE-ENDWHILE), under exact circumstances (IF-ENDIF), for a specified number of times (FOR-ENDFOR), or in specific instances (FOREACH-ENDFOR). In every case, a loop has a condition which begins the series of commands and one which ends it. Each of the commands listed here marks the beginning and end of a loop.

<u>REPEAT-UNTIL:</u>	A set of commands repeats until a condition has been satisfied.
<u>WHILE-ENDWHILE:</u>	A set of commands repeats only while a certain condition exists.
<u>FOR-ENDFOR:</u>	A set of commands repeats for a specified number of times
<u>FOREACH-ENDFOR:</u>	A set of commands repeats for a specified circumstance.
<u>IF-ENDIF:</u>	A set of commands repeats if a specified condition is satisfied.

---

## Dialog Boxes

Dialog boxes are those boxes which appear in the document window to convey important information. Most dialog boxes have buttons or list boxes to allow the user to interact with the application. You can see a good example of this when you choose Close from the File menu without having previously saved your document. A dialog box will appear in the center of your document window with a question, "Save changes to (the document name)?" followed by options indicating "Yes", "No", and "Cancel."

When you are creating macros, you can include dialog boxes to help your user interact with your macro or supply necessary information. There are four methods of getting a dialog box into your macro.

Product Commands. Many product commands are associated with dialog boxes. Any command which has the suffix of "Dlg" will bring up a dialog box (for example, *FileSaveAsDlg* will bring up the **File Save As** dialog box and *FileOpenDlg* will bring up the **File Open** dialog box). Many commands, such as the Close command, will bring up a dialog box under specific circumstances, such as when you attempt to close a document window without having first saved your document.

To define your own message and get a box that does what you want, however, you have to use one of the following three methods:

The PROMPT command. The PROMPT command creates a message box with a message that you choose and OK and Cancel buttons. It allows limited interaction between the user and the macro. If you wish greater interaction with your user, you must use a dialog box that can request and process more complex information. To do this you can use either DLL calls (calling, using, and releasing the Dynamic Link Libraries) or create a User-defined Dialog Box.

DLL commands allow you to create complex dialog boxes that look the same as any dialog boxes in the Windows environment because they have a dynamic link to the libraries of dialog box options that Windows provides. These commands are not easy to use, however, and often require careful double-checking to make sure all necessary components are included so the macro will do what it is intended to do. There is another way to get these same professional-looking results without the difficulty of DLL commands.

User-defined Dialog Box Commands. User-defined Dialog Boxes are WordPerfect's solution to remembering all the details for DLL commands. Creating a dialog box using these commands is simple because you follow the exact same format in syntax and parameters that you have been using with WordPerfect's programming and product commands. The commands then access all the dynamic links for you without you having to define all the "to" and "from" information necessary to link to these libraries.

---

## APPLICATION

Use the APPLICATION command to specify which applications will be used in the macro. This command lets you set the two-character product prefix (via the Product Prefix parameter) for product commands used with each application accessed in the macro. You may also specify the default application for any product commands in the macro that do not have a product prefix.

The APPLICATION command is executed differently than most commands; it is implemented when you compile the macro, rather than when you play the macro. This means that the APPLICATION command is always effective from its location in the macro forward, even if the APPLICATION command itself is not encountered when the macro runs. For example, if an APPLICATION command is located within an IF/ENDIF compound command, the APPLICATION command is effective from its location forward, regardless of whether the IF argument is true.

If an application listed in one of the application statements of a macro is not running, that macro will not play. Instead, a message box will display a statement telling which application caused the problem and informing the user that the macro execution is terminated.

### **syntax**

APPLICATION (*Product Prefix; Product Identifier; DEFAULT; Filename*)

The DEFAULT and Filename parameters in this command are optional. They only need to be included when you want to specify a default product prefix or a certain location for a .WCD file. So, the following commands are all valid examples of the APPLICATION command.

```
APPLICATION (WP; WPWP; Default; "wpwpus.wcd")
```

```
APPLICATION (WP; WPWP; Default)
```

```
APPLICATION (WP; WPWP; "wpwpus.wcd")
```

```
APPLICATION (WP; WPWP)
```

### **parameters**

This command has four parameters.

#### **Product Prefix**

These two letters specify the product in which following commands will run.

#### **Product Identifier**

The Product Identifier parameter specifies the application in which following commands will run. Product Identifiers reserved for WordPerfect products include the following:

<b>Product Identifier</b>	<b>Product</b>
WPWP	WordPerfect
WPSP	Speller
WPTH	Thesaurus
WPMF	Macro Facility
WPFM	File Manager

For Product Identifier names for other applications, see that application's documentation.

#### **Default**

The Default parameter works differently than most parameters in programming commands.



Rather than having a set of valid values, the Default parameter is simply used or not used. When it is used, the Default parameter makes the Product Prefix the default, so that any following product commands without Product Prefixes are treated as if they had the default Product Prefix.

**Filename**

The Filename parameter is a character expression. This parameter is optional. It specifies the name or name and location of the .WCD file containing the Product Identifier.

NEWDEFAULT

ENDAPP

Show Me!

---

## **ASSERTCANCEL**

ASSERTCANCEL causes the macro to behave as if a cancel has been encountered. If CANCELOFF has been encountered earlier in the macro, ASSERTCANCEL will have no effect.

ASSERTCANCEL is similar to RETURNCANCEL; both cause the macro to behave as if a cancel has been encountered. However, while RETURNCANCEL causes macro execution to return to the command following the parent CALL or CASE CALL command, then act as if a cancel has occurred, ASSERTCANCEL causes the macro to immediately behave as if a cancel has occurred.

### **syntax**

ASSERTCANCEL

ONCANCEL

RETURNCANCEL

Show Me!

---

## **ASSERTERROR**

ASSERTERROR causes the macro to behave as if an error has occurred. If ERROROFF has been encountered earlier in the macro, ASSERTERROR will have no effect.

ASSERTERROR is similar to RETURNERROR; both cause the macro to behave as if an error has occurred. However, RETURNERROR causes macro execution to return to the command following the parent CALL or CASE CALL command, then act as if an error has occurred, while ASSERTERROR command causes the macro to immediately behave as if an error has occurred.

### **syntax**

ASSERTERROR

ONERROR

RETURNERROR

---

## **ASSERTNOTFOUND**

ASSERTNOTFOUND causes the macro to behave as if a search has failed. ASSERTNOTFOUND is similar to RETURNNOTFOUND; both cause the macro to behave as if a search has failed. However, RETURNNOTFOUND causes macro execution to return to the command following the parent CALL or CASE CALL command, then act as if a search has failed, while ASSERTNOTFOUND causes the macro to immediately behave as if a search has failed.

### **syntax**

ASSERTNOTFOUND

ONNOTFOUND

RETURNNOTFOUND

---

## ASSIGN

ASSIGN is used to give values to WordPerfect variables. Values can be numeric, measurement, or character expressions. Two examples are given below.

ASSIGN (PLACE; "Beijing")

//Assigns the word "Beijing" to variable PLACE. Note the quotation marks around the character expression.

ASSIGN (SUM; NUMBER+19)

//Assigns the sum of 19 and the value of the variable NUMBER to the variable SUM.

### **syntax**

ASSIGN (*Variable*; *Expression*)

or

*Variable:=Expression*

### **parameters**

This command has two parameters.

#### **Variable**

Any variable is a valid value for this parameter.

#### **Expression**

Any expression is a valid value for this parameter. If the value is text enclosed in quotation marks, it must contain fewer than 512 characters.

Show Me!

---

## AboutDlg

The AboutDlg command displays the About WordPerfect dialog box, which contains the WordPerfect version number, license number information, date of program release, date of Shared DLL release, and network information, if a network is active. The AboutDlg command is the equivalent of choosing About WordPerfect from the Help menu.

### **syntax**

AboutDlg ( )

---

## Advance

Use the Advance command to position text at a certain place on a page or at a certain distance from the insertion point. This command is the equivalent of setting options in the Advance dialog box.

### **syntax**

Advance (Where:*Value Set Member*; Amount:*Measurement Expression*)

### **parameters**

This command has two parameters:

#### **Where:**

This parameter determines the direction in which text at the insertion point will be advanced, corresponding to the options in the Advance dialog box. The members and corresponding numeric equivalents of the Where value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Up!	0
Down!	1
ToLine!	2
Left!	3
Right!	4
ToPosition!	5

#### **Amount:**

This parameter is a measurement expression. If the Where parameter contains the value set member Up!, Down!, Left!, or Right!, this parameter specifies how far in that direction the text will move. If the Where parameter contains ToLine! or ToPosition!, the Advance command moves the text to that position on the page.

AdvanceDlg

---

## **AdvanceDlg**

The AdvanceDlg command displays the Advance dialog box, which lets the user position text at a certain place on a page or at a certain distance from the insertion point. This command is the equivalent of choosing Advance from the Layout menu.

### **syntax**

AdvanceDlg ( )

Advance



---

## AppMaximize

AppMaximize enlarges the WordPerfect application window to fill the desktop. If WordPerfect is already in its maximized form, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. AppMaximize is the equivalent of choosing Maximize from the WordPerfect Control menu or clicking the WordPerfect Maximize button.

WordPerfect can be reduced from its maximized form with the AppRestore command.

### **syntax**

AppMaximize ( )

ONERROR

LABEL

AppMinimize

AppRestore

DocMaximize

DocMinimize

---

## AppMinimize

AppMinimize shrinks the WordPerfect application window to an icon. If WordPerfect is already in its minimized form, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. AppMinimize is the equivalent of choosing Minimize from the WordPerfect Control menu or clicking the WordPerfect Minimize button.

WordPerfect can be expanded from its minimized form with the AppRestore command.

### **syntax**

AppMinimize ( )

LABEL

ONERROR

AppMaximize

AppRestore

DocMaximize

DocMinimize

---

## AppMove

AppMove lets the user move the WordPerfect application window to another position on the desktop. AppMove is the equivalent of choosing Move from the WordPerfect Control menu. If WordPerfect is already in its maximized form, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command.

The macro pauses when it encounters the AppMove command and the user can then use the arrow keys or mouse to move the application window. When the user clicks or presses Enter, the macro resumes from the command following the AppMove command.

### **syntax**

AppMove ( )

LABEL

ONERROR

DocMove

---

## AppRestore

AppRestore restores WordPerfect to its former size after it has been minimized or maximized. If WordPerfect has already been restored, this command terminates the macro or redirects macro execution to the LABEL command specified by ONERROR. This command is the equivalent of choosing Restore from the WordPerfect Control menu.

### **syntax**

AppRestore ( )

LABEL

ONERROR

AppMaximize

AppMinimize

DocRestore

---

## AppSize

AppSize lets the user size the WordPerfect application window. If WordPerfect is already in its maximized form, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. AppSize is the equivalent of choosing Size from the WordPerfect Control menu.

When a macro encounters AppSize, the macro pauses while the user manually sizes the WordPerfect application window. When the user clicks the mouse or presses Enter, the macro continues from the command following the AppSize command.

### **syntax**

AppSize ( )

LABEL

ONERROR

DocSize

---

## **BEEP**

The computer beeps when this command is executed.

### **syntax**

BEEP

Show Me!

---

## BYTELEN

This command assesses the number of bytes found in the Character Expression and assigns that number to the variable specified in the Variable parameter. For instance, a macro can find the number of bytes in the word "Mississippi" by using the BYTELEN command as follows.

```
BYTELEN (LN; "Mississippi")
```

This command assigns the length of the word "Mississippi" (11) to variable LN. ("Mississippi" is in quotes so the macro will recognize it as a character expression.) While standard characters are usually one byte long, extended characters may be longer than one byte. If you want to know the number of characters in a character expression, you should use the STRLEN command.

The length of the BYTELEN command is limited to a size of 512 bytes. This size includes the command, the variable, the character expression and any other characters (such as parentheses and quotation marks).

### **syntax**

BYTELEN (Variable; *Character Expression*)

### **parameters**

This command has two parameters.

#### **Variable**

Any variable is a valid value for this parameter.

#### **Character Expression**

This parameter is a character expression.

SUBBYTE

BYTEPOS

Show Me!

---

## BYTEPOS

The BYTEPOS command checks to see if the character expression specified in the Sub Expression parameter is included in the Original Expression parameter. If it is, the variable in the Variable parameter is assigned a number, corresponding to the beginning position (in bytes, not characters) of the Sub Expression parameter in the Original Expression parameter. If you want to find the character position of a substring, use the STRPOS command.

For example, to find the location of the word "Perfect" in "WordPerfect," the command would be BYTEPOS (PLACE; "Perfect"; "WordPerfect"). Variable PLACE would be given the value 5, because "Perfect" begins on the fifth byte in "WordPerfect."

If the value of the Sub Expression parameter is not found in the Original Expression parameter, the variable in the Variable parameter is given the value 0.

### **syntax**

BYTEPOS (*Variable; Sub Expression; Original Expression*)

### **parameters**

This command has three parameters.

#### **Variable**

Any variable is valid for this parameter.

#### **Sub Expression**

This parameter is a character string.

#### **Original Expression**

This parameter is a character string.

BYTELEN

SUBBYTE

Show Me!



---

## **BlockProtect**

The BlockProtect command prevents a block of selected text from being split between pages, even if that block of text would ordinarily be divided between the bottom of one page and the top of the following page. If no text is selected when the macro encounters this command, an error occurs. This command is the equivalent of choosing Page from the Layout menu, then choosing Block Protect.

### **syntax**

BlockProtect ( )

---

## **BoxEditCaption**

Use the BoxEditCaption command to edit or create a caption for a selected box. BoxEditCaption opens the Caption Editor, allowing subsequent commands to insert text and codes into the Caption Edit window. This command is the equivalent of choosing a box type from the Graphics menu, then choosing Caption.

To close the Caption window, use the Close command.

### **syntax**

BoxEditCaption ( )

BoxSelect

Close

---

## BoxEditor

The BoxEditor command opens the box editor you specify, so that a selected box may be edited. If no box is selected, then a box of the specified type is created.

### **syntax**

BoxEditor (Type:*Value Set Member*; Editor:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **Type:**

This parameter specifies which type of box to create or edit. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Figure!	0
Table!	1
Text!	2
User!	3
Equation!	4

#### **Editor:**

This parameter specifies which editor to open. If no editor is specified, the default editor for that box will be used. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Figure!	0
Text!	1
Equation!	2

BoxSelect

---

## BoxFigureEdit

The BoxFigureEdit command lets the macro perform any of the commands available in the Figure Editor. This command can retrieve and save WordPerfect graphics, as well as scale, rotate, move, mirror, invert, shadow, and outline the figures. If no figure exists or if an existing figure is not selected, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command.

### **syntax**

BoxFigureEdit (*Filename:Character Expression; ScaleX:Numeric Expression; ScaleY:Numeric Expression; Rotate:Numeric Expression; MoveX:Measurement Expression; MoveY:Measurement Expression; Mirror:Value Set Member; Invert:Value Set Member; BlackAndWhite:Value Set Member; Outline:Value Set Member; GraphOnDisk:Value Set Member; SaveFilename:Character Expression*)

### **parameters**

This command has 12 parameters:

#### **Filename:**

This character expression specifies the filename or the complete pathname of the figure to be retrieved. This parameter may be left out if no figure is to be retrieved.

#### **ScaleX:**

This numeric expression scales the width of the figure by a percentage of the original width of that figure. The value of this parameter should be from 1 to 999.

#### **ScaleY:**

This numeric expression scales the length of the figure by a percentage of the original length of that figure. The value of this parameter should be from 1 to 999.

#### **Rotate:**

This numeric expression specifies the number of degrees to rotate the figure from its original position. The value for this parameter should be from 0 to 360.

#### **MoveX:**

This measurement expression specifies the horizontal distance to move the figure from its original position. If the value is positive, the figure is moved to the right. If the value is negative, the figure is moved to the left.

#### **MoveY:**

This measurement expression specifies the vertical distance to move the figure from its original position. If the value is positive, the figure is moved up. If the value is negative, the figure is moved down.

#### **Mirror:**

This parameter specifies whether the figure will be shown as the mirror image of the original graphic image. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	The figure is shown in its original state.
Yes!	1	The figure is shown as its mirror image.

#### **Invert:**

This parameter specifies whether the colors of the figure will be inverted. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	The colors are not inverted.
Yes!	1	The colors are inverted.

**BlackAndWhite:**

This parameter specifies whether the figure will be shown in black and white instead of color. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	The figure is shown in its original colors (if you have a color monitor).
Yes!	1	The figure is shown in black and white.

**Outline:**

This parameter specifies whether only the outline of the figure will be shown. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	The figure is shown in its original state.
Yes!	1	The figure is shown as an outline.

**GraphOnDisk:**

This parameter specifies whether the modified figure will be a graphic on disk. A graphic on disk is not stored with the document; rather, it is retrieved from a separate file when the document is loaded. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	The modified figure is stored with the document.
Yes!	1	The modified figure is stored on disk.

**SaveFilename:**

This character expression specifies the path and filename used when you save the current figure. If you do not want the figure to be saved, omit this parameter. The SaveFilename parameter is the equivalent of choosing Save As from the File menu in the Figure Editor, then typing a filename for the graphic.

LABEL

ONERROR

BoxEditor

---

## BoxNewNumber

Boxes are numbered consecutively according to box type. This command causes WordPerfect to start numbering with a new number (specified in the BoxNumber parameter) from any point in the document. The new box number code is inserted at the insertion point. All boxes of the type specified in the Type parameter will be numbered starting with the new number until another box number code is encountered in the document. The BoxNewNumber command is the equivalent of choosing a box type from the Graphics menu, then choosing New Number.

### **syntax**

BoxNewNumber (Type:*Value Set Member*; BoxNumber:*Character Expression*)

### **parameters**

This command has two parameters:

#### **Type:**

This parameter specifies the type of box that the new number affects. The valid values for the Type parameter are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Figure!	0
Table!	1
Text!	2
User!	3
Equation!	4

#### **BoxNumber:**

This character expression specifies the new starting number for boxes specified in the Type parameter. Note that this parameter is a character expression, not a numeric expression. This is because boxes are not always numbered using arabic numerals--they may also be numbered using roman numerals or letters.

---

## BoxOptions

The BoxOptions command sets border, spacing, numbering, captioning, offset, and shading options for all boxes of a specified type. This command inserts a Box Options code at the insertion point. This code applies to all boxes of the specified type either to the end of the document or until a conflicting Box Option code occurs. The BoxOptions command is the equivalent of setting options in any of the Box Options dialog boxes.

### syntax

BoxOptions (Type:*Value Set Member*; LeftBorder:*Value Set Member*; RightBorder:*Value Set Member*; TopBorder:*Value Set Member*; BottomBorder:*Value Set Member*; OutLeftSpace:*Measurement Expression*; OutRightSpace:*Measurement Expression*; OutTopSpace:*Measurement Expression*; OutBottomSpace:*Measurement Expression*; InLeftSpace:*Measurement Expression*; InRightSpace:*Measurement Expression*; InTopSpace:*Measurement Expression*; InBottomSpace:*Measurement Expression*; FirstLevelNumber:*Value Set Member*; SecondLevelNumber:*Value Set Member*; CaptionPosition:*Value Set Member*; CaptionStyle:*Character Expression*; MinimumOffset:*Measurement Expression*; GrayShade:*Numeric Expression*)

### parameters

This command has 19 parameters:

#### Type:

This parameter determines the type of box that the BoxOptions command affects. The valid values for the Type value set are:

Member	Numeric Equivalent
--------	--------------------

Figure!	0
Table!	1
Text!	2
User!	3
Equation!	4

#### LeftBorder: RightBorder: TopBorder: BottomBorder:

These four parameters determine the border style for the edges of the box. The valid values for each of these parameters' value sets are:

Member	Numeric Equivalent
--------	--------------------

None!	0
Single!	1
Double!	2
Dashed!	3
Dotted!	4
Thick!	5
ExtraThick!	6

#### OutLeftSpace: OutRightSpace: OutTopSpace: OutBottomSpace:

These four measurement expressions set the outside border spacing for the edges of the box.

**InLeftSpace: InRightSpace: InTopSpace: InBottomSpace:**

These four measurement expressions set the inside border spacing for the edges of the box.

**FirstLevelNumber: SecondLevelNumber:**

These two parameters set the numbering style for the first and second level of the box. Boxes can be numbered in any combination of the options listed below; the first level specifies the number to the left of the period, the second level specifies the number to the right of the period. For example, you could specify that the first level is to be Numbers! and the second level is to be Letters! and your boxes would be sequentially numbered 1.a, 1.b, 1.c, and so forth. The valid members of these parameters' value sets are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Off!	0
Numbers!	1
Letters!	2
Roman!	3

**CaptionPosition:**

This parameter sets the caption position for the box. Types 0, 1, 2, and 3 are valid for all box types except equation boxes. Types 4, 5, 6, and 7 are valid only for equation boxes. The valid values for the CaptionPosition parameter are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

BelowOutside!	0
AboveOutside!	1
BelowInside!	2
AboveInside!	3
Below!	4
Above!	5
Left!	6
Right!	7

**CaptionStyle:**

This character expression sets the style in which the caption is numbered. For instance, the expression might be "Figure 1" or "Diagram 1-2".

**MinimumOffset:**

This measurement expression sets the minimum distance required from the top of the paragraph to the graphic. This specifies how far the box can move up into the paragraph when WordPerfect needs to reduce the offset measurement.

**GrayShade:**

This numeric expression specifies the percentage of gray shading the box should have. The value of this parameter should be from 0 to 100, with 0 being no shading, and 100 being black.



---

## BoxPosition

You can use the BoxPosition command to change the anchor type, box type, height, position, and width for a selected graphics box. You can also specify whether text should wrap around the box. The BoxPosition command is the equivalent of setting options in any of the Box Position dialog boxes.

### syntax

BoxPosition (NewType:Value Set Member; Anchor:Value Set Member; SkipPages:Numeric Expression; AutoMode:Value Set Member; Width:Measurement Expression; Height:Measurement Expression; VerticalType:Value Set Member; VerticalPosition:Measurement Expression; HorizontalType:Value Set Member; HorizontalPosition:Measurement Expression; ColumnStart:Numeric Expression; ColumnEnd:Numeric Expression; WrapText:Value Set Member)

### parameters

This command has 13 parameters:

#### NewType:

This parameter changes the selected box to the specified box type. If you don't want to change the box type, omit this parameter. The valid values for the NewType value set are:

Member	Numeric Equivalent
--------	--------------------

Figure!	0
Table!	1
Text!	2
User!	3
Equation!	4

#### Anchor:

This parameter sets the anchor type for a selected box. The valid values for the Anchor value set are:

Member	Numeric Equivalent
--------	--------------------

Paragraph!	0
Page!	1
Character!	2

#### SkipPages:

This numeric expression specifies the number of pages to skip before inserting the box. The SkipPages parameter should be an integer. This parameter is the equivalent of typing a number in the Number of Pages to Skip box. This is only valid for page-anchored boxes.

#### AutoMode:

This parameter specifies whether the width and/or height of the box are automatically sized by WordPerfect, corresponding to the options in the pop-up list in the Size group box. The members and corresponding numeric equivalents for the Automode value set are:

Member	Numeric Equivalent
--------	--------------------

AutoBoth!	0
-----------	---

AutoWidth!	1
AutoHeight!	2
SetBoth!	3

**Width: Height:**

These measurement expressions set the width and height, respectively, of the selected box.

**VerticalType:**

This parameter sets the beginning vertical point for the selected box. The valid values and anchor types for the VerticalType value set are:

Member	Numeric Equivalent	Valid Anchor Types
FullPage!	0	Page
Top!	1	Character, Page
Center!	2	Character, Page
Bottom!	3	Character, Page
Baseline!	4	Character
Specify!	5	Page, Paragraph

**VerticalPosition:**

This measurement expression is used to set the exact beginning vertical position for the selected box, when the VerticalType parameter is set to Specify!. If the VerticalType parameter is set to anything but Specify!, the VerticalPosition parameter may be omitted.

This measurement expression is only valid for page-anchored and paragraph-anchored boxes.

**HorizontalType:**

This parameter sets the beginning horizontal point for the selected box. The valid values and anchor types for the HorizontalType value set are:

Member	Numeric Equivalent	Valid Anchor Types
Left!	0	Page, Paragraph
Right!	1	Page, Paragraph
Center!	2	Page, Paragraph
Full!	3	Page, Paragraph
ColumnLeft!	4	Page
ColumnRight!	5	Page
ColumnCenter!	6	Page
ColumnFull!	7	Page
Specify!	8	Page

**HorizontalPosition:**

This measurement expression is used to set the exact beginning horizontal position for the selected box, when the HorizontalType parameter is set to Specify!.

This measurement expression is only valid for page-anchored and paragraph-anchored

boxes.

**ColumnStart:**

The value of this numeric expression should be an integer. The ColumnStart parameter is used when the selected box should span across two or more columns. This parameter specifies the leftmost column the selected box should occupy.

This numeric expression is valid only for a page-anchored box with a horizontal column placement.

**ColumnEnd:**

The value of this numeric expression should be an integer. The ColumnEnd parameter is used when the selected box should span across two or more columns. This parameter specifies the rightmost column the selected box should occupy.

This numeric expression is valid only for a page-anchored box with a horizontal column placement.

**WrapText:**

This parameter specifies whether text in the document should wrap around the selected figure, or write over it. The members and corresponding numeric expressions for this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	Text will ignore the box and continue printing from margin to margin.
Yes!	1	Text will wrap around the selected box.

BoxSelect

---

## BoxRetrieve

BoxRetrieve creates a box at the insertion point. The box type is specified in the Type parameter. This command is the equivalent of choosing Figure from the Graphics menu, then choosing Retrieve.

The BoxRetrieve command also lets you specify a file to retrieve. The specified file must be a graphics file.

### **syntax**

BoxRetrieve (Type:*Value Set Member*; Filename:*Character Expression*)

### **parameters**

This command has two parameters:

#### **Type:**

This parameter specifies the box type to be created. The members and corresponding numeric equivalents of the Type value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Figure!	0
---------	---

Table!	1
--------	---

Text!	2
-------	---

User!	3
-------	---

Equation!	4
-----------	---

#### **Filename:**

This parameter must contain the filename of a graphics file. The Filename parameter may contain just the file name, or the entire pathname.

If the box being created does not use graphics, this parameter should be left blank.

BoxSelect

BoxEditor

BoxFigureEdit

---

## BoxSelect

The BoxSelect command selects a box of a specified type and number, allowing subsequent commands to modify that box.

### **syntax**

BoxSelect (Type:*Value Set Member*; BoxNumber:*Character Expression*)

### **parameters**

This command has two parameters:

#### **Type:**

This parameter specifies the type of box to be selected. The valid values for the Type value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Figure!	0
Table!	1
Text!	2
User!	3
Equation!	4

#### **BoxNumber:**

This character expression specifies the number of the box to be selected. If BoxNumber is not specified, the BoxSelect command will select the next box in the document. Note that this parameter is a character expression, not a numeric expression. This is because box numbers are not always expressed as arabic numerals--they may also be expressed as roman numerals or letters.

---

## ButtonBarAddItem

The ButtonBarAddItem command adds a button to the current Button Bar to execute the macro specified in the MacroName parameter.

### **syntax**

ButtonBarAddItem (ButtonText: *Character Expression*; MacroName: *Character Expression*; BitMapLink: *Numeric Expression*; BitMapID: *Numeric Expression*)

### **parameters**

This command has four parameters.

#### **ButtonText**

This parameter is a character expression. The ButtonText is the text that appears on the button. If you wish to have this text appear in the button, use #999 as the BitMapID parameter.

#### **MacroName**

This parameter is a character expression. The MacroName specifies the name of the macro that will play when the user selects this button from the Button Bar.

#### **BitMapLink**

This parameter is a numeric expression. The BitMapLink is the hInstance of the module in which the bit map is located. If you have chosen to use a WordPerfect bitmap, you can find the hInstance of the bitmap by finding the name of the token, then looking in the .WCD file where the token name is listed. The number you need for this parameter is listed next to the token name.

#### **BitMapID**

This parameter is a numeric expression. The BitMapID is the number of the bitmap you have chosen to use. If you are using a WordPerfect bitmap, this parameter should be given the numeric expression of 0.

[ButtonBarDeleteItem](#)

---

## ButtonBarDeleteltem

The ButtonBarDeleteltem command deletes a button from the Button Bar that executes the macro specified in the MacroName parameter.

### **syntax**

ButtonBarDeleteltem (MacroName: *Character Expression*)

### **parameters**

This command has one parameter.

#### **MacroName**

This parameter is a character expression. The MacroName is the name of the macro that will be deleted from the Button Bar.

[ButtonBarAddItem](#)

---

## ButtonBarEditDlg

The ButtonBarEditDlg command opens the Button Bar Edit dialog box, which lets the user add buttons or macros to the Button Bar. This command is the equivalent of choosing Button Bar Setup from the View menu, then choosing Edit.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

ButtonBarEditDlg ( )

ButtonBarNewDlg

ButtonBarOptionsDlg

ButtonBarSelectDlg



---

## ButtonBarNewDlg

The ButtonBarNewDlg creates a blank [Button Bar](#) and opens the Edit Button Bar [dialog box](#), which lets the user create a new Button Bar. This command is the equivalent of choosing Button Bar Setup from the View [menu](#), then choosing New.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

ButtonBarNewDlg ( )

[ButtonBarEditDlg](#)

[ButtonBarOptionsDlg](#)

[ButtonBarSaveDlg](#)

[ButtonBarSelectDlg](#)

---

## ButtonBarOptions

The ButtonBarOptions command specifies whether the Button Bar will display on the top, bottom, left, or right side of the WordPerfect workspace, and whether each button will display with a picture, with text, or with both. The ButtonBarOptions command is the equivalent of setting options in the Button Bar Options dialog box.

### **syntax**

ButtonBarOptions (Style:Value Set Member; Location:Value Set Member)

### **parameters**

This command has two parameters:

#### **Style:**

This parameter specifies how the buttons in the Button Bar will appear. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

PictureOnly!	0
TextOnly!	1
PictureAndText!	2

#### **Location:**

This parameter specifies where the Button Bar displays in the document window. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

BBarOnTop!	0
BBarOnLeft!	1
BBarOnRight!	2
BBarOnBottom!	3

ButtonBarOptionsDlg

---

## ButtonBarOptionsDlg

The ButtonBarOptionsDlg command opens the Button Bar Options dialog box, which lets the user specify whether the Button Bar will display on the top, bottom, left, or right side of the WordPerfect workspace, and whether each button will display with a picture, with text, or with both. This command is the equivalent of choosing Button Bar Setup from the View menu, then choosing Options.

### **syntax**

ButtonBarOptionsDlg ( )

ButtonBarNewDlg

ButtonBarSelectDlg

ButtonBarEditDlg

---

## ButtonBarSave

The ButtonBarSave command saves the current Button Bar under a specified filename. This command is the equivalent of saving a Button Bar in the Save Button Bar dialog box.

### **syntax**

ButtonBarSave (Filename:*Character Expression*)

### **parameters**

This command has one parameter:

#### **Filename:**

This character expression specifies the name under which the Button Bar is to be saved. The Filename parameter may contain just the filename, or the entire pathname.

ButtonBarSaveDlg

---

## ButtonBarSaveDlg

The ButtonBarSaveDlg command opens the Save Button Bar dialog box, which lets the user save the active Button Bar. This command is the equivalent of choosing Button Bar Setup from the View menu, then choosing Save As.

### **syntax**

ButtonBarSaveDlg ( )

ButtonBarNewDlg

ButtonBarSelectDlg

ButtonBarOptionsDlg

ButtonBarEditDlg

---

## ButtonBarSelect

The ButtonBarSelect command selects and displays a specified Button Bar. This command is the equivalent of choosing a Button Bar file in the Select Button Bar dialog box.

### **syntax**

ButtonBarSelect (Filename:*Character Expression*)

### **parameters**

This command has one parameter:

#### **Filename:**

This character expression specifies the Button Bar to be selected. The Filename parameter may contain just the file name, or the entire pathname.

ButtonBarSelectDlg

---

## ButtonBarSelectDlg

The ButtonBarSelectDlg command opens the Select Button Bar dialog box, which lets the user select a previously created Button Bar file to open. This command is the equivalent of choosing Button Bar Setup from the View menu, then choosing Select.

### **syntax**

ButtonBarSelectDlg ( )

ButtonBarNewDlg

ButtonBarOptionsDlg

ButtonBarSaveDlg

ButtonBarEditDlg

---

## ButtonBarShow

Use the ButtonBarShow command to show or hide the Button Bar. This command is the equivalent of choosing Button Bar from the View menu to show or hide the Button Bar.

### **syntax**

ButtonBarShow (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether the Button Bar is shown or hidden. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
Off!	0	The Button Bar is hidden.
On!	1	The Button Bar is shown.



---

## **//Comment**

The Comment command (//) tells the WordPerfect Macro Compiler to ignore all text between the command (//) and the next hard return. The maximum length of any comment is 512 bytes

You can use this command to indicate the function of a certain part of a macro, to state the types of values a particular variable might contain, or to include any other information that may be helpful.

The Comment command can be on a line by itself or it can follow another command on the same line.

### **syntax**

//Comment

Show Me!

---

## **CALL**

The CALL command sends (branches) macro execution to the LABEL command specified in the Label parameter. The macro runs from that point until it encounters a RETURN command. Macro execution then returns to the command following the CALL command and resumes from that point.

### **syntax**

CALL (*Label*@)

### **parameters**

This command has one parameter.

### **Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

CASE

RETURN

Show Me!

---

## **CANCELOFF**

CANCELOFF causes Cancel commands from the keyboard to be ignored while the macro is running. However, you can still cancel the macro using the Macro Facility menu.

**syntax**  
CANCELOFF

CANCELON

Show Me!

---

## **CANCELON**

CANCELON allows Cancel commands from the keyboard to be accepted while the macro is running. If the macro encounters an ONCANCEL command before Cancel is entered from the keyboard, execution is sent to the LABEL command specified in the ONCANCEL command. If no ONCANCEL command has been encountered, the macro will stop. The macro default is CANCELON (Cancel commands are accepted), so you will not need to use this command unless you use a CANCELOFF command earlier in the macro.

### **syntax**

CANCELON

CANCELOFF

LABEL

ONCANCEL

Show Me!

---

## CASE

The CASE command specifies which part of the macro will run next. There are three parts to this command: the Test, the Cases and Labels, and the Default Label.

When the macro runs, the CASE command first finds the value of the Test parameter and checks that value against the value of each of the Case parameters. If a Case parameter has the same value as the Test parameter, the macro goes to the LABEL command specified after that Case parameter. If the Test parameter doesn't match any of the Case parameters, the macro goes to the LABEL command specified by the Default Label parameter.

For example, when a CASE command looks like this:

```
CASE (YESNO; {"Y";yes@;"y ";yes@"N";no@"n";no@};maybe@)
```

and variable YESNO (the Test parameter) has the value "y" or "Y", the macro jumps to the LABEL (yes@) command. If variable YESNO has the value "n" or "N", the macro jumps to the LABEL (no@) command. If variable YESNO doesn't equal either of the case expressions, the macro jumps to the LABEL (maybe@) command.

### **syntax**

*CASE (Test; {Case; Label@; Case; Label@; . . . Case; Label@}Default Label@)*

### **parameters**

This command has several parameters.

#### **Test**

The Test parameter is usually a variable, but may be any expression.

#### **Case**

Any number of Case parameters may be between the braces ({}). Case parameters are usually words or numbers, but can be expressions. Each case must be followed by a Label parameter.

#### **Label**

Any number of Label parameters may be between the braces ({}). Each label specified must correspond to a LABEL command within the macro. Case and Label parameters within the braces should always come in pairs, with the Label parameter following the Case parameter.

#### **Default Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

### CASE CALL

Show Me!

---

## CASE CALL

The CASE CALL command specifies which LABEL command in the macro to jump to next. The macro runs from that LABEL command until the macro encounters a RETURN command. The macro then runs from the command immediately below the CASE CALL command.

There are three parts to this command: the Test, the Cases and Labels, and the Default Label. When the macro runs, the CASE CALL command first finds the value of the Test parameter, and checks that value against the value of each of the Case parameters. If a Case parameter has the same value as the Test parameter, the macro runs from the LABEL command specified after the Case parameter. If the Test parameter doesn't match any of the Case parameters, the macro goes to the LABEL command specified by the Default Label parameter.

For example, when a CASE CALL command looks like this:

```
CASE CALL(YESNO; {"Y";yes@;"y"; yes@;"N";no@;"n"; no@}; maybe@)
```

and variable YESNO (the Test parameter) has the value "y" or "Y", the macro jumps to the LABEL (yes@) command. If variable YESNO has the value "n" or "N", the macro jumps to the LABEL (no@) command. If variable YESNO doesn't equal either of the case expressions, the macro jumps to the LABEL (maybe@) command. In any of these instances, the macro continues running from the specified LABEL command until it encounters a RETURN command. The macro then continues at the command immediately below the CASE CALL command.

### **syntax**

```
CASE CALL (Test; {Case; Label@; Case; Label@; . . . Case; Label@};Default Label@)
```

### **parameters**

This command has several parameters.

#### **Test**

The Test parameter is usually a variable, but may be any expression.

#### **Case**

Any number of Case parameters may be between the braces ( { } ). Case parameters are usually words or numbers, but can be expressions. Each Case parameter is a potential match for the Test parameter. When there is a match, macro execution is directed to the LABEL command specified by the Label parameter for that case. Each case must be followed by a Label parameter.

#### **Label**

Any number of Label parameters may be between the braces ( { } ). Each label specified must correspond to a LABEL command within the macro. Case and Label parameters within the braces should always come in pairs, with the Label parameter following the Case parameter.

#### **Default Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

CASE  
CALL  
RETURN

Show Me!

---

## CHAIN

The CHAIN command runs another specified macro when the current macro is completed, regardless of where the CHAIN command is located in the macro. If a macro contains more than one CHAIN command, only the last CHAIN command has an effect. A QUIT command at the end of the active macro overrides any pending CHAIN command.

The macro this command runs must already be compiled. This means that you must have already played the macro once, or you must have compiled the macro using the Macro Facility.

### **syntax**

CHAIN (*Macro Name*)

### **parameters**

This command has one parameter.

#### **Macro Name**

Any character expression. This parameter should be the full pathname of an existing macro.

RUN

Show Me!



---

## CaptionBoxNumber

The CaptionBoxNumber command inserts the current caption box number into the caption editor at the insertion point. If a caption editor is not active when this command is encountered, an error occurs and the macro is terminated or macro execution is redirected to the LABEL command specified by ONERROR. The CaptionBoxNumber command is the equivalent of choosing Box Number in any caption editor.

### **syntax**

CaptionBoxNumber ( )

LABEL

ONERROR

Close

---

## CaseToLower

The CaseToLower command converts any selected text to lowercase. If no text is selected, an error occurs and the macro is terminated or macro execution is redirected to the LABEL command specified by ONERROR. The CaseToLower command is the equivalent of choosing Convert Case from the Edit menu, then choosing Lowercase.

### **syntax**

CaseToLower ( )

LABEL

ONERROR

CaseToUpper

---

## CaseToUpper

The CaseToUpper command converts any selected text to uppercase. If no text is selected, an error occurs and the macro is terminated or macro execution is redirected to the LABEL command specified by ONERROR. The CaseToUpper command is the equivalent of choosing Convert Case from the Edit menu, then choosing Uppercase.

### **syntax**

CaseToUpper ( )

LABEL

ONERROR

CaseToLower

---

## Close

The Close command closes any active window or text editor. This command is the equivalent of choosing Close from the File menu.

If you want a macro to close a document without saving it, use the CloseNoSave command before the Close command.

### **syntax**

Close (*Save:Value Set Member*)

### **parameters**

This command has one parameter:

#### **Save:**

This parameter specifies whether changes you have made should be saved. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	The changes are discarded.
Yes!	1	The changes are kept.

CloseNoSave

---

## CloseNoSave

Use the CloseNoSave command to close the active document without saving it or closing the document window. For instance, if Document 2 is active when the CloseNoSave command is encountered, it is closed without being saved. The Document 2 window (now empty) is still active.

Note that this command does not delete any copies of the document on disk; it just discards the copy in the active document window.

### **syntax**

CloseNoSave (Verify:Value Set Member)

### **parameters**

This command has one parameter:

#### **Verify:**

This parameter specifies whether the macro prompts the user to close the active document. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	The user is not prompted; the document is closed.
Yes!	1	The user is asked whether to close the active document.

Close

---

## ColumnDefine

The ColumnDefine command inserts a column definition code at the top of the paragraph in the active document if Auto Code Placement is on. If Auto Code Placement is off, the column definition code is inserted at the insertion point in the active document. This command lets you define the column type, the number of columns, and the column margins. The ColumnDefine command is the equivalent of setting options in the Define Columns dialog box.

If a ColumnDefine command is used while in columns mode, a new [Col Def] code is placed just before the active [Col On] code. If an existing [Col Def] code is encountered in front of the [Col On] code, that [Col Def] code will be replaced.

Use this command for inserting column codes when you need to specify the margins of each column. If you want columns of even width, use the ColumnDefineEven command.

### syntax

ColumnDefine (NumberOfColumns:*Numeric Expression*; Type:*Value Set Member*;  
ColumnsOn:*Value Set Member*; LeftMargin1:*Measurement Expression*;  
RightMargin1:*Measurement Expression*; LeftMargin2:*Measurement Expression*;  
RightMargin2:*Measurement Expression*; . . .LeftMargin24:*Measurement Expression*;  
RightMargin24:*Measurement Expression*)

### parameters

This command has 51 parameters:

#### NumberOfColumns:

This numeric expression specifies the number of columns for the column definition. The value of this parameter should be an integer from 2 to 24 (24 is the maximum number of columns available on a page).

#### Type:

This parameter specifies the type of columns to be used in the column definition. The valid values of the Type value set are:

Member	Numeric Equivalent
--------	--------------------

Newspaper!	0
Parallel!	1
ParallelWithBlockProtect!	2

#### ColumnsOn:

This parameter specifies whether columns will be turned on immediately following the column definition. The default is Off!. The valid values for this parameter are:

Member	Numeric Equivalent
--------	--------------------

Off!	0
On!	1

#### LeftMargin1: RightMargin1: LeftMargin2: RightMargin2: . . .LeftMargin24: RightMargin24:

These measurement expressions specify the margins for each column. There should be a LeftMargin and RightMargin parameter for each column. For instance, if the

NumberOfColumns parameter has a value of 3, there should be three pairs of LeftMargin and RightMargin parameters, so the command might appear as:

```
ColumnDefine(NumberOfColumns:3; Type:Parallel!; ColumnsOn:On!; LeftMargin1:0.75i;  
RightMargin1:1.5i; LeftMargin2:1.9i; RightMargin2:4.0i; LeftMargin3:4.3i; RightMargin3:7.5i)
```

WordPerfect allows 24 columns per page, so the LeftMargin and RightMargin parameters should not exceed that value. Each successive LeftMargin and RightMargin parameter should have a greater value than the previous parameter (although the left margin may equal the previous right margin). Otherwise, the columns will overlap.

ColumnDefineDlg

ColumnDefineEven

ColumnsOn

---

## ColumnDefineDlg

The ColumnDefineDlg command opens the Define Columns dialog box, which lets the user specify the column type, the number of columns, and column margins. This command is the equivalent of choosing Columns from the Layout menu, then choosing Define.

### **syntax**

ColumnDefineDlg ( )

ColumnDefine

ColumnDefineEven



---

## ColumnDefineEven

The ColumnDefineEven command inserts a column definition code at the top of the paragraph in the active document if Auto Code Placement is on. If Auto Code Placement is off, the column definition code is inserted at the insertion point in the active document. This command lets you define the column type, number of columns, and number of margins. The ColumnDefineEven command is the equivalent of setting options in the Define Columns dialog box.

Use this command to insert column codes when you want columns that are the same width. If you need to specify the margins of each column, use the ColumnDefine command.

### **syntax**

ColumnDefineEven (NumberOfColumns:*Numeric Expression*; Type:*Value Set Member*; ColumnsOn:*Value Set Member*)

### **parameters**

This command has 3 parameters:

#### **NumberOfColumns:**

This numeric expression specifies the number of columns for the column definition. The value of this parameter should be an integer from 2 to 24 (24 is the maximum possible number of columns available on a page).

#### **Type:**

This parameter specifies the type of columns to be used in the column definition. The valid values of the Type value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Newspaper!	0
Parallel!	1
ParallelWithBlockProtect!	2

#### **ColumnsOn:**

This parameter specifies whether columns will be turned on immediately following the column definition. The valid values for this parameter are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
On!	1

ColumnDefine

ColumnDefineDlg

ColumnsOn

---

## ColumnsOff

The ColumnsOff command turns off the column definition that is active at the insertion point. If columns are not on at the insertion point, this command will return an error and the macro will terminate or macro execution will be redirected to the LABEL command specified by ONERROR. This command is the equivalent of choosing Columns from the Layout menu, then choosing Columns Off.

### **syntax**

ColumnsOff ( )

LABEL

ONERROR

ColumnDefine

ColumnDefineEven

ColumnsOn

---

## ColumnsOn

The ColumnsOn command turns on the preceding column definition and inserts a [Col On] code at the top of the paragraph in the active document if Auto Code Placement is on. If Auto Code Placement is off, a [Col On] code is inserted at the insertion point of the active document. If no columns have been defined at the point where the command is encountered, or if columns are already on at that point, an error is returned and the macro is terminated or macro execution is redirected to the LABEL command specified by ONERROR.. The ColumnsOn command is the equivalent of choosing Columns from the Layout menu, then choosing Columns On.

### **syntax**

ColumnsOn ( )

LABEL

ONERROR

ColumnDefine

ColumnDefineEven

ColumnsOff

---

## CommentConvertToText

The CommentConvertToText command converts the comment preceding insertion point to text. If no comments precede the insertion point, this command converts the comment following the insertion point to text. If no comments exist in the active document, macro execution is terminated or is redirected to the LABEL command specified by an ONERROR. This command is the equivalent of choosing Comment from the Tools menu, then choosing Convert to Text.

### **syntax**

CommentConvertToText ( )

LABEL

ONERROR

CommentCreate

CommentEditDlg

---

## CommentCreate

Use the CommentCreate command to create comments and replace existing comments. This command lets you specify the text to be used in the comment. The CommentCreate command is the equivalent of choosing Comment from the Tools menu, then choosing Create or Edit.

### **syntax**

CommentCreate (CommentText:*Character Expression*; Replace:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **CommentText:**

This character expression is the text that will appear in the comment.

#### **Replace:**

This parameter specifies whether the text specified in the CommentText parameter will replace an existing comment or be inserted into a new comment. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

WordPerfect has a certain method for determining which comment to replace. From the insertion point, WordPerfect searches backward until it finds a comment to replace. If no comment precedes the insertion point, the first comment in the document is replaced. If there are no comments in the document, a comment is created at the insertion point.

[CommentConvertToText](#)

[CommentEditDlg](#)

[CommentShow](#)

---

## CommentCreateDlg

The CommentCreateDlg command displays the Create Comment dialog box, which lets the user type the text for a comment that will be placed at the insertion point in the active document. This is the equivalent of choosing Comment from the Tools menu, then choosing Create.

### **syntax**

CommentCreateDlg ( )

CommentConvertToText

CommentCreate

CommentEditDlg

---

## CommentEditDlg

The CommentEditDlg command displays the Edit Comment dialog box, which lets the user edit the text of an existing comment.

WordPerfect uses a certain method to search for the comment to be edited. From the insertion point, WordPerfect first searches backwards. If it finds a comment, that comment is displayed in the Edit Comment dialog box. If no comment is found when searching backward, WordPerfect edits the first comment in the document. If no comment is found in the document, an error is returned and the macro is terminated or macro execution is redirected to the LABEL command specified by ONERROR.

### **syntax**

CommentEditDlg ( )

LABEL

ONERROR

CommentConvertToText

CommentCreate

CommentCreateDlg

---

## CommentShow

Use the CommentShow command to show or hide comments in a document. This command is the equivalent of choosing Comment from the View menu to show or hide comments.

### **syntax**

CommentShow (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is not specified, the command acts as a "toggle."

### **State:**

This parameter determines whether the comments are shown. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
Off!	0	Comments are hidden.
On!	1	Comments are shown.



---

## ConditionalEOP

The ConditionalEOP command inserts a code at the insertion point in the active document that keeps a specified number of lines from being separated by a page break. If a page break would ordinarily occur in that group of lines, the page will instead break before the line on which the conditional end of page code is located.

### **syntax**

ConditionalEOP (KeepTogether:*Numeric Expression*)

### **parameters**

This command has one parameter:

#### **KeepTogether:**

This numeric expression specifies how many lines should be kept together following the Conditional EOP code. The KeepTogether parameter should be an integer.

ConditionalEOPDlg

---

## ConditionalEOPDIg

The ConditionalEOPDIg command displays the Conditional End of Page dialog box, which lets the user specify the number of lines to keep together on one page in a document. The code inserted at the insertion point in the active document will keep that number of lines from being separated by a page break. The ConditionalEOPDIg command is the equivalent of choosing Page from the Layout menu, then choosing Conditional End of Page.

### **syntax**

ConditionalEOPDIg ( )

ConditionalEOP

---

## ConvertClipboardDlg

The ConvertClipboardDlg command brings up the Convert Windows Picture dialog box and offers three options for saving/convertng a Windows Metafile (.WMI). This command is the equivalent of choosing Preferences from the File menu, then choosing Graphics Import.

This command is not recordable. To use this command, you must edit your macro.

### **syntax**

ConvertClipboardDlg ( )

ConvertClipboardPicture

---

## ConvertClipboardPicture

Use the ConvertClipboardPicture command to import pictures in the following formats: Windows format (.WMF), WordPerfect format (.WPG), or both. This command is the equivalent of choosing Graphics Import from Preferences on the File menu.

### **syntax**

ConvertClipboardPicture (ConvertTo:*Value Set Member* )

### **parameters**

This command has one parameter:

#### **ConvertTo**

This parameter specifies whether the picture will be saved in Windows format (metafile), converted to WordPerfect format, or both.

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Metafile!	0
WPG!	1
Both!	2

ConvertClipboardDlg

---

## CrossRefMark

Use the CrossRefMark command to create a reference in one part of the document that will help a user to find information in another part of the document. The reference is an indicator (such as "see page 238") which directs the user to another part of the document (the target of the reference). This is useful for referring to a chart or page number in a document. This command is the equivalent of setting options in the Mark Cross-Reference dialog box.

### **syntax**

CrossRefMark (TargetName:*Character Expression*; MarkType:*Value Set Member*; RefType:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **TargetName:**

This character expression specifies the name of the cross-reference. This parameter is the equivalent of typing text in the Target Name box in the Mark Cross-Reference dialog box.

#### **MarkType:**

This parameter specifies whether this command will place a marker for the reference, the target, or both. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

TargetOnly!	0
RefOnly!	1
BothMode!	2

#### **RefType:**

This parameter specifies what type of object the reference should be tied to. This parameter is not relevant when marking for TargetOnly!. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

PageNumber!	0
ParagraphOutline!	1
FootnoteNumber!	2
EndnoteNumber!	3
FigureNumber!	4
TableBoxNumber!	5
TextBoxNumber!	6
UserBoxNumber!	7
EquationBoxNumber!	8

CrossRefMarkDlg

---

## CrossRefMarkDlg

The CrossRefMarkDlg command displays the Mark Cross-Reference dialog box, which lets the user specify what should be cross referenced, what the reference should be tied to, and the target name of the reference. The CrossRefMarkDlg command causes the macro to pause until the user presses Enter or Cancel. This command is the equivalent of choosing Mark Text from the Tools menu, then choosing Cross-Reference.

### **syntax**

CrossRefMarkDlg ( )

CrossRefMark

---

## **DEFAULTUNITS**

The DEFAULTUNITS command specifies a unit of measure to be used with measurement expressions that don't contain a Unit of Measure parameter. Once this command has been encountered, any measurement string or expression without a specified unit of measure will use the unit of measure specified in the DEFAULTUNITS command.

The unit of measure specified in the DEFAULTUNITS command is effective until another DEFAULTUNITS command specifying a different unit of measure is encountered.

### **syntax**

DEFAULTUNITS (*Unit of Measure*)

### **parameters**

This command has one parameter:

#### **Unit of Measure**

This parameter has four valid values. These words are not character strings or expressions, so they do not need to be enclosed within quotes. Since the values are not character strings, a variable cannot be used as the parameter. The valid values are as follows:

centimeters

inches

points

wpunits ("wpunits" stands for WordPerfect units; one WordPerfect unit equals 1/1200".)

[Show Me!](#)

---

## DLLCALL

**CAUTION:** The DLLCALL, DLLLOAD, and DLLFREE statements deal with programming code outside of WordPerfect. If used incorrectly, these commands can damage files and/or reset your computer. Do not use these commands unless you are familiar with programming in Windows and using Dynamic Link Libraries (DLLs).

**Note:** The data type descriptions that follow assume the reader has a working knowledge of the Microsoft Windows 3.x SDK and Microsoft C data types. All functions called in a DLL via the DLLCALL command must be declared as FAR PASCAL.

DLLCALL accesses a function of the Dynamic Link Library (DLL) specified in a corresponding DLLLOAD command. The variable in the Link parameter of this command must contain a value which it was assigned in the DLLLOAD command. This value, which is a Windows handle to the DLL, links the macro to the DLL specified in the DLL parameter of the DLLLOAD command. The Function parameter specifies which function from the DLL will be invoked. The information in the To DLL parameter is passed to that function. The function processes the information, then returns a value to the variable specified in the From DLL parameter.

If a DLL writes data into a variable, the Address operator must be used with the expression. Failure to use this operator may cause the DLL to overwrite memory and cause a UAE or other undesirable result. Passing an expression without the use of the Address operator is acceptable only when the DLLCALL will only read the data. NOTE that the largest amount of data that can be written into the variable is 1K bytes.

### **syntax**

DLLCALL (*Link; Function; From DLL:Data Type; {To DLL}*)

Note the unusual syntax structure for this command. Instead of the usual semicolon, a colon (:) separates the From DLL and Data Type parameters.

### **parameters**

This command has five parameters:

#### **Link**

This parameter is a variable. The value for this variable must be established via the DLLLOAD command before the DLLCALL command is encountered.

#### **Function**

This parameter is a character expression. The Function parameter determines which function in the DLL specified by the Link parameter will be accessed by the macro. The name of this function depends on the DLL being accessed; refer to the DLL's documentation for the Function parameter's valid values.

#### **From DLL**

This parameter is a variable. The From DLL parameter is assigned a value from the function specified by the Function parameter in the DLL specified by the Link parameter. The type of value this variable expects to receive depends on the value of the Data Type parameter (see Data Type below).

#### **Data Type**

This parameter specifies the data type of the information that is being returned by the DLL function into the variable specified in the From DLL parameter. It is critically important that this data type be compatible with the data type that will actually be returned from the function. Failure to do so will most likely result in an "Unrecoverable Application Error" or system crash.

---

#### **Data Type**

#### **Equivalent Data Type**

---



BOOL	Windows SDK data type BOOL.
DWORD	Windows SDK data type DWORD.
INTEGER	Windows SDK data type LONG
REAL	Microsoft C type double.
STRING	Windows SDK data type LPSTR.
WORD	Windows SDK data type WORD.

### **To DLL**

This parameter (or series of parameters) is the information passed to the DLL function. This parameter may be empty (if so, it will appear as a pair of braces), contain a single parameter between the braces, or contain a series of parameters between the braces, each separated by a semicolon (;). In any case, an open brace ({) begins this parameter set and a close brace (}) ends the parameter set. Individual parameters within this parameter set have the following format:

#### **<expression>**

The value of <expression> will be passed directly into the DLL function. If <expression> results in an integer, the function must expect a LONG or DWORD for this parameter. If <expression> results in a string, the function must expect an LPSTR for this parameter. If <expression> results in a Boolean, the function must expect a BOOL for this parameter.

#### **HIWORD ( <expression> )**

The value of <expression> must result in an integer. The high order 16 bits of this 32 bit integer will be passed to the DLL function. The function must expect an int, short, or WORD for this parameter.

#### **LOWORD ( <expression> )**

The value of <expression> must result in an integer. The low order 16 bits of this 32 bit integer will be passed to the DLL function. The function must expect an int, short, or WORD for this parameter.

#### **ADDRESS ( <variable> )**

A far pointer to the address of the contents of the variable will be passed to the DLL function. If the variable contains an integer, the function must expect an LPDWORD. If the variable contains a Boolean, the function must expect a BOOL FAR \*. If the variable contains a string, the function must expect an LPSTR.

DLLFREE

DLLLOAD

User-Defined Dialog Boxes

---

## **DLLFREE**

**CAUTION:** The DLLCALL, DLLLOAD, and DLLFREE statements deal with programming code outside of WordPerfect. If used incorrectly, these commands can damage files and/or reset your computer. Do not use these commands unless you are familiar with programming in Windows and using Dynamic Link Libraries (DLLs).

The DLLFREE command breaks the links between the macro and Dynamic Link Library (DLL) which were established with the DLLLOAD command. After the DLLFREE command has been encountered, the variable specified in the Link parameter no longer corresponds to the DLL specified in the DLL parameter in the DLLLoad command. The value of the variable specified in the Link parameter is left undefined.

The link between the macro and the DLL must be broken before the macro terminates. The most efficient way to do this is to use the DLLFREE command. However, the macro interpreter also tracks all DLLs loaded with the DLLLOAD command and frees any that weren't freed with a DLLFREE command when the execution of a macro terminates.

### **syntax**

DLLFREE (*Link*)

### **parameters**

This command has one parameter:

#### **Link**

This parameter is a variable. The DLLFREE command does not assign a value to this variable.

DLLCALL

DLLLOAD

User-defined Dialog Boxes

---

## DLLLOAD

**CAUTION:** The DLLCALL, DLLLOAD, and DLLFREE statements deal with programming code outside of WordPerfect. If used incorrectly, these commands can damage files and/or reset your computer. Do not use these commands unless you are familiar with programming in Windows and using Dynamic Link Libraries (DLLs).

The DLLLOAD command establishes a link between the macro and the Dynamic Link Library (DLL) specified in the DLL parameter. This command assigns a numeric value (the Windows handle to the DLL) to the variable specified in the Link parameter. This variable can then be used in the DLLCALL command to access a function of the DLL specified in the DLL parameter.

The link between the macro and the DLL must be broken before the macro terminates. The most efficient way to do this is to use the DLLFREE command. However, the macro interpreter also tracks all DLLs loaded with the DLLLOAD command and frees any that weren't freed with a DLLFREE command when the execution of a macro terminates.

### **syntax**

DLLLOAD (*Link*; *DLL*)

### **parameters**

This command has two parameters:

#### **Link**

This parameter is a variable. The value assigned to this variable is the Windows handle to the DLL and establishes the connection between the macro and the DLL specified in the DLL parameter.

#### **DLL**

This parameter is a character expression that specifies the file's name or the complete path of the DLL to be accessed by the macro.

DLLFREE

DLLCALL

User-Defined Dialog Boxes

---

## DateCode

The DateCode command inserts the current date in your document as a code. The date is updated each time you retrieve or print that document. This command is the equivalent of choosing Date from the Tools menu, then choosing Code.

### **syntax**

DateCode ( )

DateFormat

DateText

---

## DateFormat

The DateFormat command specifies the way the Date feature formats the date. This command is similar to setting options in the Document Date/Time Format [dialog box](#).

### syntax

DateFormat (DefinitionString:*Character Expression*)

### parameters

This command has one parameter:

#### DefinitionString:

This character expression specifies how the dates are displayed when you use the Date feature. However, the method of formatting the date is different than in the Document Date/Time Format dialog box.

In the list below, the characters on the left represent ways the parts of the date and time can be written. You can include words, spaces, and other punctuation with these numbers to create a date format.

Character	Meaning
1	Day of the Month
2	Month (number)
3	Month (word)
4	Year (all four digits)
5	Year (two digits)
6	Day of the Week (word)
7	Hour (24-hour clock)
8	Hour (12-hour clock)
9	Minute
0	am/pm
&	Second
%	Used before a number. A percent sign (%) will pad numbers less than 10 with a leading zero, or abbreviate the month or day of the week.
\$	Used before a number. A dollar sign (\$) will pad numbers less than 10 with a leading space.

#### Examples:

3 1, 4 = January 1, 1992  
%6 %3 1, 4 = Wed Jan 1, 1992  
%2/%1/5 (6) = 01/01/92 (Wednesday)  
\$2/\$1/5 (%6) = 1/ 1/92 (Wed)  
8:90 = 10:55am

#### DateCode

DateFormatDlg

DateText

PrefDateFormat

---

## **DateFormatDlg**

This command displays the Date/Time Format dialog box, which lets you specify how the date and time are displayed when you insert them into your document using the Date feature. This command is the equivalent of choosing Date from the Tools menu, then choosing Format.

### **syntax**

DateFormatDlg ( )

---

## **DateText**

The DateText command inserts the current date as text, which can then be edited. This command is the equivalent of choosing Date from the Tools menu, then choosing Text.

### **syntax**

DateText ( )

DateCode

DateFormat



---

## DDECreateLink

The DDECreateLink command establishes a link between a WordPerfect document and a file in another application, so that as data changes in the source file, the changes appear in the WordPerfect document as well. This command is the equivalent of setting options in the Create DDE Link dialog box.

### **syntax**

DDECreateLink (LinkName:*Character Expression*; Source:*Character Expression*;  
UpdateMode:*Value Set Member*; StoragePreference:*Value Set Member*)

### **parameters**

This command has four parameters:

#### **LinkName:**

This character expression identifies the link. It is the name used to represent the link in the DDE Link dialog boxes and in the DDE Link codes shown in the Reveal Codes window. This parameter corresponds to the Link Name box in the Create DDE Link dialog box.

#### **Source:**

This character expression specifies the application, the file and, optionally, an item within that file that is being linked to the WordPerfect document. The application and filenames are separated by a vertical bar (|). If an item (a section of the file) is specified, a second vertical bar separates the item from the filename.

This parameter corresponds to the Source File and Item box in the Create DDE Link dialog box.

#### **UpdateMode:**

This parameter specifies whether the DDE link will be updated automatically or manually, corresponding to the radio buttons in the Update Mode group box in the Create DDE Link dialog box. The default is Automatic!. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Manual!	0
Automatic!	1

#### **StoragePreference:**

This parameter specifies how to represent the data in the WordPerfect document, and corresponds to the radio buttons in the Storage Type box. The default is Text!. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Graphics!	0
Text!	1

DDECreateLinkDlg

DDEPasteLink

---

## **DDECreateLinkDlg**

The DDECreateLinkDlg command displays the Create DDE Link dialog box, which lets the user set up a Dynamic Data Exchange (DDE) link between a WordPerfect document and a file in another application. This command is the equivalent of choosing Links from the Edit menu, then choosing Create.

### **syntax**

DDECreateLinkDlg ( )

DDECreateLink

---

## DDEDeleteLink

The DDEDeleteLink command removes from 1 to 20 specified links. This command is the equivalent of deleting links in the Delete [DDE Link](#) dialog box.

### **syntax**

DDEDeleteLink (LinkName:Character Expression; LinkName2:Character Expression; LinkName3:Character Expression; LinkName4:Character Expression; LinkName5:Character Expression; LinkName6:Character Expression; LinkName7:Character Expression; LinkName8:Character Expression; LinkName9:Character Expression; LinkName10:Character Expression; LinkName11:Character Expression; LinkName12:Character Expression; LinkName13:Character Expression; LinkName14:Character Expression; LinkName15:Character Expression; LinkName16:Character Expression; LinkName17:Character Expression; LinkName18:Character Expression; LinkName19:Character Expression; LinkName20:Character Expression)

### **parameters**

This command has 20 parameters. You only need to include as many of the parameters as you have links to delete.

**LinkName: LinkName2: LinkName3: LinkName4: LinkName5: LinkName6:  
LinkName7: LinkName8: LinkName9: LinkName10: LinkName11: LinkName12:  
LinkName13: LinkName14: LinkName15: LinkName16: LinkName17: LinkName18:  
LinkName19: LinkName20:**

These parameters are each character expressions. Each LinkName parameter specifies the name of a link to be deleted.

[DDEDeleteLinkDlg](#)

---

## **DDEDeleteLinkDlg**

The DDEDeleteLinkDlg command displays the Delete DDE Link dialog box, which lets the user delete Dynamic Data Exchange (DDE) links. This command is the equivalent of choosing Links from the Edit menu, then choosing Delete.

### **syntax**

DDEDeleteLinkDlg ( )

DDEDeleteLink

---

## DDEEditLink

The DDEEditLink command changes the properties of an existing Dynamic Data Exchange (DDE) link. This command is the equivalent of setting options in the Edit DDE Link dialog box.

### **syntax**

DDEEditLink (LinkName:*Character Expression*; NewName:*Character Expression*;  
NewSource:*Character Expression*; NewUpdateMode:*Value Set Member*)

### **parameters**

This command has four parameters:

#### **LinkName:**

This character expression identifies the link to be changed. This parameter corresponds to the Link Name box in the Edit DDE Link dialog box.

#### **NewName:**

This character expression changes the name of the link to the name specified in this parameter. This parameter corresponds to the New Link Name box in the Edit DDE Link dialog box. If this parameter is not included, the link name will not be changed.

#### **NewSource:**

This character expression specifies the application, the file and, optionally, an item within that file to which the link in the WordPerfect document will be changed. Vertical bars (|) are used to separate the application, file, and item names. This parameter corresponds to the Source File and Item box in the Edit DDE Link dialog box.

If this parameter is not included, the source of the link will not be changed.

#### **NewUpdateMode:**

This parameter specifies whether the DDE link will be updated automatically or manually, corresponding to the radio buttons in the Update group box in the Edit DDE Link dialog box. The valid values are:

---

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Manual!	0
---------	---

Automatic!	1
------------	---

If this parameter is missing, the link's update mode will not be changed.

DDECreateLink

DDEEditLinkDlg

---

## **DDEEditLinkDlg**

The DDEEditLinkDlg command displays the Edit DDE Link dialog box, which lets the user change the properties of existing Dynamic Data Exchange (DDE) links. This command is the equivalent of choosing Links from the Edit menu, then choosing Edit.

### **syntax**

DDEEditLinkDlg ( )

DDECreateLink

DDEEditLink

---

## **DDEPasteLink**

The DDEPasteLink command creates a link between a WordPerfect document and a file in another application. Information that has been copied to the Clipboard from the other application and is then pasted into WordPerfect using the DDEPasteLink command will be automatically updated in the WordPerfect document whenever the information is changed in the source file. This command is the equivalent of choosing Links from the Edit menu, then choosing Paste Link.

WordPerfect determines the source of the link from the information on the Clipboard. WordPerfect then creates a name to represent the link in the DDE Link dialog boxes and in the Reveal Codes window, and updates the link automatically whenever the source file changes. To control these settings while creating a link, use the DDECreateLink command rather than DDEPasteLink.

### **syntax**

DDEPasteLink ( )

DDECreateLink

---

## DDEUpdateLink

The DDEUpdateLink command updates all specified Dynamic Data Exchange (DDE) links between WordPerfect and other Windows applications. This command is equivalent to updating links in the Update Link [dialog box](#).

### **syntax**

DDEUpdateLink (LinkName:Character Expression; LinkName2:Character Expression; LinkName3:Character Expression; LinkName4:Character Expression; LinkName5:Character Expression; LinkName6:Character Expression; LinkName7:Character Expression; LinkName8:Character Expression; LinkName9:Character Expression; LinkName10:Character Expression; LinkName11:Character Expression; LinkName12:Character Expression; LinkName13:Character Expression; LinkName14:Character Expression; LinkName15:Character Expression; LinkName16:Character Expression; LinkName17:Character Expression; LinkName18:Character Expression; LinkName19:Character Expression; LinkName20:Character Expression)

### **parameters**

This command has 20 parameters:

**LinkName: LinkName2: LinkName3: LinkName4: LinkName5: LinkName6:  
LinkName7: LinkName8: LinkName9: LinkName10: LinkName11: LinkName12:  
LinkName13: LinkName14: LinkName15: LinkName16: LinkName17: LinkName18:  
LinkName19: LinkName20:**

Each of these character expressions specifies the name of a link to be updated. You only need to use as many of these parameters as you have links to update.

[DDEUpdateLinkDlg](#)



---

## **DDEUpdateLinkDlg**

The DDEUpdateLinkDlg command displays the Update DDE Link dialog box, which lets the user update Dynamic Data Exchange (DDE) links between WordPerfect and other Windows applications. The DDEUpdateLinkDlg command is the equivalent of choosing Links from the Edit menu, then choosing Update.

### **syntax**

DDEUpdateLinkDlg ( )

DDEUpdateLink

---

## DeleteCharNext

The DeleteCharNext command deletes the character to the right of the insertion point. This command is the equivalent of pressing Delete.

### **syntax**

DeleteCharNext ( )

DeleteCharPrevious

---

## DeleteCharPrevious

The DeleteCharPrevious command deletes the character to the left of the insertion point. This command is the equivalent of pressing Backspace.

### **syntax**

DeleteCharPrevious ( )

DeleteCharNext

---

## DeleteEOL

The DeleteEOL command deletes all text and codes from the insertion point to the end of the current text line. The DeleteEOL command is the equivalent of pressing Ctrl+Del.

### **syntax**

DeleteEOL ( )

---

## DeleteEOP

The DeleteEOP command deletes all text and codes from the insertion point to the end of the current page. The DeleteEOP command is the equivalent of pressing Ctrl+Shft+Del.

### **syntax**

DeleteEOP ( )

---

## DeleteToBeginningOfWord

The DeleteToBeginningOfWord command deletes all text and codes from the insertion point to the beginning of the current word (the insertion point must be positioned within a word for this command to take effect).

### **syntax**

DeleteToBeginningOfWord ( )

DeleteToEndOfWord

---

## DeleteToEndOfWord

The DeleteToEndOfWord command deletes all text and codes from the insertion point to the beginning of the following word.

### **syntax**

DeleteToEndOfWord ( )

DeleteToBeginningOfWord

---

## DeleteWord

This command deletes the word at the insertion point. If the insertion point is between words, this command deletes the word the insertion point is touching.

### **syntax**

DeleteWord ( )



---

## Display

The Display command specifies whether changes made by the macro are displayed on the screen while the macro is running. When Display is on, the user can see all the changes the macro makes, but the macro slows down considerably. When display is off, the macro is faster, but the user cannot see the changes the macro is making until the macro ends.

The default display state is off, so if you wish to have Display set to "on", your macro must contain this command.

When the display state is off, the "Please Wait" message appears in the status line. When the display state is on, the message does not appear, but macro execution is displayed.

### **syntax**

Display (State:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **State:**

This parameter specifies whether the changes the macro makes to documents will be shown while the macro is running. The values of this member set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

DisplayFontAdjustment

InhibitInput

Show Me!

---

## DisplayFontAdjustment

The DisplayFontAdjustment command adjusts the screen position of text and codes according to the currently selected font. This command does not take effect if WordPerfect is in Draft mode and does not affect how the document will print. This command is the equivalent of setting Display Font Adjustment options in the Display Pitch dialog box.

### **syntax**

DisplayFontAdjustment (Mode:*Value Set Member*; PercentOfNormal:*Numeric Expression*)

### **parameters**

This command has two parameters:

#### **Mode:**

This parameter specifies whether font adjustment will be set to auto or manual, corresponding to the Display Font Adjustment options in the Display Pitch dialog box. The valid members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Auto!	0
-------	---

Manual!	1
---------	---

#### **PercentOfNormal:**

This numeric expression specifies the percentage of the characters' actual size the display should show. The PercentOfNormal parameter is only necessary if the Mode parameter is set to Manual!. If the Mode parameter is set to Auto!, this parameter is ignored. The value of this parameter should be a positive integer from 5 to 250.

DisplayPitch

---

## DisplayPitch

The DisplayPitch command sets the amount of horizontal space between each character when WordPerfect is in draft mode. This command is the equivalent of setting Draft Mode options in the Display Pitch dialog box.

### **syntax**

DisplayPitch (Mode:*Value Set Member*; Pitch:*Measurement Expression*)

### **parameters**

This command has two parameters:

#### **Mode:**

This parameter specifies whether display pitch will be set to auto or manual, corresponding to the Draft Mode options in the Display Pitch dialog box. The valid members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Auto!	0
-------	---

Manual!	1
---------	---

#### **Pitch:**

This parameter is a measurement expression. The Pitch parameter specifies the amount of horizontal space allotted to each character. The valid range of values is 0.025" to 0.500". The Pitch parameter is only necessary if the Mode parameter is set to Manual!. If the Mode parameter is set to Auto!, the Pitch parameter is ignored.

DisplayFontAdjustment

DisplayPitchDlg

---

## DisplayPitchDlg

The DisplayPitchDlg command displays the Display Pitch dialog box, which lets the user set the amount of horizontal space between characters in draft mode. This command is the equivalent of choosing Document from the Layout menu, then choosing Display Pitch.

### **syntax**

DisplayPitchDlg ( )

DisplayPitch

---

## DocCompare

The DocCompare command compares the active document to a specified document saved on disk. WordPerfect notes any differences between the two documents by redlining additions to the active document and adding strikeout text to indicate text that has been removed from the active document. Text that has been moved to a different location is also marked. This command is the equivalent of choosing Document Compare from the Tools menu, specifying a filename in the Add Markings dialog box, then choosing Compare.

### **syntax**

DocCompare (ComparisonFile:*Character Expression*)

### **parameters**

This command has one parameter:

#### **ComparisonFile:**

This character expression specifies the filename of the document to which the active document is compared. If this parameter is omitted, DocCompare attempts to compare the current document with its copy on disk.

DocCompareRemove

---

## **DocCompareAddMarkingsDlg**

The DocCompareAddMarkingsDlg command displays the Add Markings dialog box, which lets the user specify which document on disk the active document will be compared to. This command is the equivalent of choosing Document Compare from the Tools menu, then choosing Add Markings.

### **syntax**

DocCompareAddMarkingsDlg ( )

DocCompareRemoveMarkingsDlg

---

## DocCompareRemove

The DocCompareRemove command removes markings that were created by the Document Compare feature. The DocCompareRemove command is the equivalent of setting the Leave Redline Marks option in the Remove Markings [dialog box](#), then choosing OK.

### **syntax**

DocCompareRemove (KeepRedline:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **KeepRedline:**

This parameter specifies whether redline marks should remain in the document comparison (strikeout is removed regardless). The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

[DocCompare](#)

---

## **DocCompareRemoveMarkingsDlg**

The DocCompareRemoveMarkingsDlg command displays the Remove Markings dialog box, which lets the user specify whether to remove the Redline marks inserted during the document compare. This command is the equivalent of choosing Document Compare from the Tools menu, then choosing Remove Markings.

### **syntax**

DocCompareRemoveMarkingsDlg ( )

DocCompare

DocCompareAddMarkingsDlg



---

## **DocInitialCodes**

The DocInitialCodes command opens the Document Initial Codes window for the active document. The Document Initial Codes window lets you specify default codes for the active document.

Once a macro has encountered this command, following commands will be inserted into the Document Initial Codes window. The Close command closes the Document Initial Codes window, and makes the document window active again.

### **syntax**

DocInitialCodes ( )

Close

---

## DocInitialFont

The DocInitialFont command sets the default font for the active document. This command is the equivalent of setting a typeface and size in the Initial Font dialog box.

### **syntax**

DocInitialFont (Name:*Character Expression*; Size:*Numeric Expression*)

### **parameters**

This command has two parameters:

#### **Name:**

This character expression specifies the typeface to be used as the document's initial font. This parameter should correspond to a font available on the selected printer.

#### **Size:**

This numeric expression specifies the height of the initial font in points (72 points equal one inch). This parameter should only be used with printer drivers that have scalable fonts available. If the currently selected printer driver does not have scalable fonts, this parameter should be left blank.

DocInitialFontDlg

---

## **DocInitialFontDlg**

The DocInitialFontDlg command displays the Document Initial Font dialog box, which lets the user select an initial typeface and font height for the active document. This command is the equivalent of choosing Document from the Layout menu, then choosing Initial Font.

### **syntax**

DocInitialFontDlg ( )

DocInitialFont

---

## DocMaximize

The DocMaximize command enlarges the active document to fill the entire screen. If the document is already maximized, macro execution is terminated or is redirected to the LABEL command specified by an ONERROR. This command is the equivalent of choosing Maximize from the document control menu.

### **syntax**

DocMaximize ( )

LABEL

ONERROR

AppMaximize

DocMinimize

---

## DocMinimize

The DocMinimize command shrinks the active document to an icon. If the document is already minimized, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. This command is the equivalent of choosing Minimize from the document control menu.

### **syntax**

DocMinimize ( )

LABEL

ONERROR

AppMinimize

DocMaximize

---

## **DocMove**

The DocMove command lets the user move the active document to another position. This command is the equivalent of choosing Move from the document's control menu.

### **syntax**

DocMove ( )

AppMove

---

## **DocNext**

The DocNext command cycles the active document to the back, and the other documents forward, making the next document active. The method used by Windows to order multiple documents is to layer each new document on top of the current one. So when you use DocNext, the "next" document is the one that was most recently open before you opened the current document. If only one document is open, this command has no effect. This command is the equivalent of choosing Next from the document's control menu.

### **syntax**

DocNext ( )

DocPrevious

---

## **DocPrevious**

If Windows 3.1 is the current operating environment, the DocPrevious command brings the backmost document window to the front and activates it. If Windows 3.0 is the current operating environment, this command works the same as the DocNext command . If only one document is open, this command has no effect.

### **syntax**

DocPrevious ( )

DocNext



---

## DocRedlineMethod

The DocRedlineMethod command specifies how redlined text will appear on the printed page. This command is the equivalent of setting options in the Document Redline dialog box under Document in the Layout menu.

### **syntax**

DocRedlineMethod (Method:*Value Set Member*; Character:*Character Expression*)

### **parameters**

This command has two parameters:

#### **Method:**

The Method parameter specifies how redline markings will appear, corresponding to the radio buttons in the Redline Method dialog box under Document in the Layout menu: Printer Dependent, Mark Left Margin, or Mark Alternating Margins. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

PrinterDependent!	0
Left!	1
Alternating!	2

#### **Character:**

This character expression specifies the character to be placed in the margin of redline text when the Method parameter is specified as Left! or Alternating!. If the Method parameter is specified as PrinterDependent!, the Character parameter should be left blank.

DocRedlineMethodDlg

---

## **DocRedlineMethodDlg**

The DocRedlineMethodDlg command displays the Document Redline dialog box, which lets the user set options for how redline text will appear on the printed page. The DocRedlineMethodDlg command is the equivalent of choosing Document from the Layout menu, then choosing Redline Method.

### **syntax**

DocRedlineMethodDlg ( )

DocRedlineMethod

---

## DocRestore

DocRestore restores the active document to its former size after having been minimized or maximized. If the document has already been restored, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. This command is the equivalent of choosing Restore from the document Control menu.

### **syntax**

DocRestore ( )

LABEL

ONERROR

AppRestore

DocMaximize

DocMinimize

---

## DocSize

DocSize lets the user size the active document window. DocSize is the equivalent of choosing Size from the WordPerfect Control menu.

When a macro encounters DocSize, the macro pauses while the user manually sizes the window. When the user clicks the mouse or presses Enter, the macro continues from the command following the DocSize command. If the active document window is maximized, an error is returned and macro execution is terminated or redirected to the LABEL specified by an ONERROR command.

### **syntax**

DocSize ( )

LABEL

ONERROR

AppSize

---

## DocSummaryDefine

The DocSummaryDefine command specifies the information for the document summary for the active document window. This command is the equivalent of typing information in the text boxes in the Document Summary dialog box.

### **syntax**

DocSummaryDefine (Name:*Character Expression*; Type:*Character Expression*;  
Date:*Character Expression*; Author:*Character Expression*; Typist:*Character Expression*;  
Subject:*Character Expression*; Account:*Character Expression*; Keywords:*Character  
Expression*; Abstract:*Character Expression*)

### **parameters**

This command has nine parameters. Unless otherwise specified, if a parameter is omitted, that part of the Document Summary is not changed. For example, if the Name parameter is omitted and no descriptive name already exists for the document, that part of the document summary is simply left blank.

#### **Name:**

This character expression is the equivalent of filling in the Descriptive Name text box in the Document Summary dialog box.

#### **Type:**

This character expression is the equivalent of filling in the Descriptive Type text box in the Document Summary dialog box.

#### **Date:**

This character expression is the equivalent of filling in the Creation Date text box in the Document Summary dialog box. If you omit this parameter when defining a new document summary, the current date automatically becomes the creation date.

#### **Author:**

This character expression is the equivalent of filling in the Author text box in the Document Summary dialog box.

#### **Typist:**

This character expression is the equivalent of filling in the Typist text box in the Document Summary dialog box.

#### **Subject:**

This character expression is the equivalent of filling in the Subject text box in the Document Summary dialog box.

#### **Account:**

This character expression is the equivalent of filling in the Account text box in the Document Summary dialog box.

#### **Keywords:**

This character expression is the equivalent of filling in the Keywords text box in the Document Summary dialog box.

#### **Abstract:**

This character expression is the equivalent of filling in the Abstract text box in the Document Summary dialog box.

DocSummaryDlg

DocSummaryDelete



---

## **DocSummaryDelete**

The DocSummaryDelete command deletes all document summary information for the active document. This command is the equivalent of choosing Delete in the Document Summary dialog box, then choosing Yes in the confirmation box.

### **syntax**

DocSummaryDelete ( )

DocSummaryDefine

DocSummaryDlg

---

## **DocSummaryDlg**

The DocSummaryDlg command displays the Document Summary dialog box, which lets the user create and save general information about a document. This command is the equivalent of choosing Document from the Layout menu, then choosing Summary.

### **syntax**

DocSummaryDlg ( )

DocSummaryDefine

DocSummaryDelete



---

## **DocSummaryExtract**

The DocSummaryExtract command gathers certain information in a document, such as the author's and typist's name, subject matter, and a document abstract, and puts this information into a Document Summary. This command is the equivalent of choosing Extract in the Document Summary dialog box.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

DocSummaryExtract ( )

DocSummaryDefine

DocSummaryDlg

---

## DocSummaryGetData

Use the DocSummaryGetData command to assign information from a document summary field to a specified variable. The macro can then use that variable in conjunction with other commands to decide whether to perform certain actions.

### **syntax**

DocSummaryGetData (Name:*Variable*; Type:*Variable*; Date:*Variable*; Author:*Variable*; Typist:*Variable*; Subject:*Variable*; Account:*Variable*; Keywords:*Variable*; Abstract:*Variable*)

### **parameters**

This command has nine parameters. All of these parameters are optional.

#### **Name:**

The information contained in the text box of the Descriptive Name text box in the Document Summary dialog box is assigned to this variable.

Any variable name is a valid value for this parameter.

#### **Type:**

The information contained in the text box of the Descriptive Type text box in the Document Summary dialog box is assigned to this variable.

Any variable name is a valid value for this parameter.

#### **Date:**

The information contained in the text box of the Creation Date text box in the Document Summary dialog box is assigned to this variable.

Any variable name is a valid value for this parameter.

#### **Author:**

The information contained in the text box of the Author text box in the Document Summary dialog box is assigned to this variable.

Any variable name is a valid value for this parameter.

#### **Typist:**

The information contained in the text box of the Typist text box in the Document Summary dialog box is assigned to this variable.

Any variable name is a valid value for this parameter.

#### **Subject:**

The information contained in the text box of the Subject text box in the Document Summary dialog box is assigned to this variable.

Any variable name is a valid value for this parameter.

#### **Account:**

The information contained in the text box of the Account text box in the Document Summary dialog box is assigned to this variable.

Any variable name is a valid value for this parameter.

#### **Keywords:**

The information contained in the text box of the Keywords text box in the Document Summary dialog box is assigned to this variable.

Any variable name is a valid value for this parameter.

#### **Abstract:**

The information contained in the text box of the Abstract text box in the Document

Summary dialog box is assigned to this variable.  
Any variable name is a valid value for this parameter.

DocSummaryDlg

---

## **DocSummaryPrint**

The DocSummaryPrint command prints the document summary for the active document. This command is the equivalent of choosing Print in the Document Summary dialog box.

### **syntax**

DocSummaryPrint ( )

DocSummaryDlg

---

## DocSummarySaveAs

The DocSummarySaveAs command specifies the information for the document summary for the active document and saves that information under a specified filename. This command is the equivalent of typing information in the text boxes in the Document Summary dialog box, choosing Save As, then specifying a filename in the Save Document Summary dialog box.

### **syntax**

DocSummarySaveAs (Filename:*Character Expression*; DocName:*Character Expression*; DocType:*Character Expression*; CreationDate:*Character Expression*; Author:*Character Expression*; Typist:*Character Expression*; Subject:*Character Expression*; Account:*Character Expression*; Keywords:*Character Expression*; Abstract:*Character Expression*)

### **parameters**

This command has 10 parameters. Unless otherwise specified, if a parameter is omitted, that part of the Document Summary is not saved. For example, if the Name parameter is omitted and no descriptive name already exists for the document, that part of the document summary is simply left blank.

#### **Filename:**

This character expression specifies the name of the file under which the document summary information is saved. This parameter is the equivalent of typing a filename in the Save As text box in the Save Document Summary dialog box. Do not omit this parameter.

#### **DocName:**

This character expression is the equivalent of filling in the Descriptive Name text box in the Document Summary dialog box.

#### **DocType:**

This character expression is the equivalent of filling in the Descriptive Type text box in the Document Summary dialog box.

#### **CreationDate:**

This character expression is the equivalent of filling in the Creation Date text box in the Document Summary dialog box. If you omit this parameter when defining a new document summary, the creation date automatically becomes the new date.

#### **Author:**

This character expression is the equivalent of filling in the Author text box in the Document Summary dialog box.

#### **Typist:**

This character expression is the equivalent of filling in the Typist text box in the Document Summary dialog box.

#### **Subject:**

This character expression is the equivalent of filling in the Subject text box in the Document Summary dialog box.

#### **Account:**

This character expression is the equivalent of filling in the Account text box in the Document Summary dialog box.

#### **Keywords:**

This character expression is the equivalent of filling in the Keywords text box in the Document Summary dialog box.

#### **Abstract:**

This character expression is the equivalent of filling in the Abstract text box in the Document Summary dialog box.

---

## DraftMode

The DraftMode command can be used to make WordPerfect text display as mono-spaced characters (like the DOS version of WordPerfect). This command is the equivalent of choosing Draft Mode from the View menu.

### **syntax**

DraftMode (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is not included, the command acts as a "toggle."

### **State:**

This parameter specifies whether WordPerfect displays text in draft mode. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
Off!	0	WordPerfect will not display text in draft mode.
On!	1	WordPerfect will display text in draft mode

---

## **ELSE**

The ELSE command can be used in conjunction with the IF command to specify what the macro should do when the IF statement is false. Commands placed between the ELSE and the ENDIF commands are executed only if the value of the relational expression in the IF statement is false.

If no ELSE command is given in an IF conditional statement, when the IF statement is false the macro continues at the command below the ENDIF statement. It may help to think of the ELSE command as meaning "otherwise". For instance, a macro may ask if a copy of a file has been made. If the answer is yes (true), the macro continues to play. Otherwise, the macro will prompt the user to save a copy of the document.

### **syntax**

ELSE

IF

ENDIF

Show Me!



---

## **ENDAPP**

The ENDAPP command indicates that an application specified in a previous APPLICATION command will no longer be used in the current macro. Any product commands with the application's Product Prefix following this command will not be valid.

The ENDAPP command is executed differently than most commands; it is implemented when you compile the macro, rather than when you play the macro. This means that the ENDAPP command is always effective from its location in the macro, even if the ENDAPP command itself is not encountered when the macro runs.

### **syntax**

ENDAPP(Product Prefix)

### **parameters**

This command has one parameter:

#### **Product Prefix**

This parameter is two letters. The Product Prefix parameter must correspond to a Product Prefix set in an APPLICATION command earlier in the macro. This parameter is not a character expression, so the letters should not be enclosed in quotation marks.

### APPLICATION

---

## **ENDFOR**

ENDFOR is the closing command for the FOR-ENDFOR conditional loop and the FOREACH-ENDFOR loop. ENDFOR is used to indicate the end of FOR and FOREACH loops.

### **syntax**

ENDFOR

FOR

FOREACH

Show Me!

---

## **ENDIF**

ENDIF is the closing command for the IF-ENDIF conditional statement. ENDIF is used to indicate the end of IF conditional statements.

### **syntax**

ENDIF

IF

ELSE

Show Me!

---

## **ENDPROMPT**

The ENDPROMPT command closes the floating message box created by the PROMPT command.

**syntax**  
ENDPROMPT

PROMPT  
MacroStatusPrompt

Show Me!

---

## **ENDWHILE**

ENDWHILE is the closing command for the WHILE-ENDWHILE loop. A Test expression is evaluated at the top of the WHILE-ENDWHILE loop. If the expression is false, the WHILE-ENDWHILE loop will not execute, and macro execution resumes with the command following the ENDWHILE command. For information on conditional loops whose text expressions are evaluated at the end of the loop, see REPEAT and UNTIL.

### **syntax**

ENDWHILE

WHILE

Show Me!

---

## **ERROROFF**

The ERROROFF command causes application errors to be ignored while the macro is running. However, macro errors, such as when the syntax is incorrect, cannot be ignored.

**syntax**  
ERROROFF

ERRORON

---

## **ERRORON**

The ERRORON command allows application errors to occur while the macro is running. Macro execution will then be sent to the LABEL command specified in the most recently encountered ONERROR command, or if no ONERROR command has been encountered, the macro will stop. The macro default is ERRORON (errors are allowed), so you will not need to use this command unless an ERROROFF command occurs earlier in the macro.

### **syntax**

ERRORON

ERROROFF

---

## **EditAppend**

The EditAppend command appends selected text or graphics to the end of the contents already in the Clipboard. This command is the equivalent of choosing Append from the Edit menu.

### **syntax**

EditAppend ( )



---

## **EditCopy**

The EditCopy command copies the selected text or graphics to the Clipboard, replacing any items that may already be on the Clipboard. This command is the equivalent of choosing Copy from the Edit menu. If no text or graphic is selected, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command.

### **syntax**

EditCopy ( )

LABEL

ONERROR

---

## **EditCut**

The EditCut command removes selected text and graphics from the document window and puts them on the Clipboard, replacing any items that may already be on the Clipboard. This command is the equivalent of choosing Cut from the Edit menu. If no text or graphic is selected, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command.

### **syntax**

EditCut ( )

LABEL

ONERROR

---

## **EditPaste**

The EditPaste command pastes the contents of the Clipboard at the insertion point in the active document. If text or graphics are selected when the EditPaste command is encountered, the selected items are replaced with the contents of the Clipboard. This command is the equivalent of choosing Paste from the Edit menu.

### **syntax**

EditPaste ( )

---

## EndnoteCreate

The EndnoteCreate command opens an Endnote window. Following commands can then insert text and codes into this window. When the endnote is completed, a Close command can close the Endnote window, placing a reference to the endnote at the insertion point in the active document. The EndnoteCreate command is the equivalent of choosing Endnote from the Layout menu, then choosing Create.

### **syntax**

EndnoteCreate ( )

Close

---

## EndnoteEdit

The EndnoteEdit command opens the Endnote window for the endnote you specify. The Endnote window contains the text and codes for the endnote. Once this window is open, all macro commands apply to the Endnote window until the Close command is encountered. The EndnoteEdit command is the equivalent of choosing Endnote from the Layout menu, then choosing Edit.

### **syntax**

EndnoteEdit (EndnoteNumber:*Character Expression*)

### **parameters**

This command has one parameter. If no parameter is specified, the endnote following the insertion point is edited.

#### **EndnoteNumber:**

This character expression specifies the number of the endnote to be edited. Note that this parameter is a character expression, not a numeric expression. This is because endnote numbers are not always expressed as arabic numerals; they may also be expressed as roman numerals or characters.

Close

EndnoteEditDlg

---

## EndnoteEditDlg

The EndnoteEditDlg command displays the Edit Endnote dialog box, which lets the user specify which endnote to edit in the active document. This command is the equivalent of choosing Endnote from the Layout menu, then choosing Edit.

### **syntax**

EndnoteEditDlg ( )

EndnoteEdit

---

## EndnoteNewNumber

The EndnoteNewNumber command inserts a code at the insertion point which begins new numbering for all following endnotes. This command is the equivalent of entering a number or character in the New Endnote Number dialog box.

### **syntax**

EndnoteNewNumber (EndnoteNumber:*Character Expression*)

### **parameters**

This command has one parameter:

#### **EndnoteNumber:**

This character expression specifies the new starting number or character for endnote numbers. Note that this parameter is a character expression, not a numeric expression. This is because endnote numbers are not always expressed as arabic numerals; they may also be expressed as roman numerals or characters.

EndnoteNewNumberDlg

---

## EndnoteNewNumberDlg

The EndnoteNewNumberDlg command displays the New Endnote Number dialog box, which lets the user set a number from which following endnotes will be incremented. This command is the equivalent of choosing Endnote from the Layout menu, then choosing New Number.

### **syntax**

EndnoteNewNumberDlg ( )

EndnoteNewNumber



---

## EndnoteOptions

The EndnoteOptions command sets options for all endnotes following the [insertion point](#). These options include numbering style, amount of space between endnotes, and the length an endnote must be before it can span a page break. The EndnoteOptions command is the equivalent of setting options in the Endnote Options [dialog box](#) found by selecting Options under Endnote on the Layout menu.

### syntax

EndnoteOptions (NumberingMethod:*Value Set Member*; Characters:*Character Expression*; StyleText:*Character Expression*; StyleNote:*Character Expression*; LineSpacing:*Numeric Expression*; BetweenSpacing:*Measurement Expression*; MinimumHeight:*Measurement Expression*)

### parameters

This command has seven parameters:

#### NumberingMethod:

This parameter specifies what will be used as the endnote markers: numbers, letters, or characters. The NumberingMethod value set has three valid members, corresponding to the items in the Numbering Method pop-up list in the Endnote Options dialog box:

Member	Numeric Equivalent
Numbers!	0
Characters!	1
Letters!	2

#### Characters

This parameter is necessary only when the Style parameter of this command is set to Characters!. The Characters parameter specifies which character(s) to use in endnote markers. This character expression should be no more than five characters long.

#### StyleText:

This character expression determines how the endnote number appears in the text of the document. This information must be entered in the Style in Text box in the Endnote Options dialog box while recording a macro. This inserts an "embedded" code into the StyleText character expression, which can be viewed using [Reveal Codes](#). If you want to insert an embedded code into an existing StyleText parameter in a macro, record another macro containing that code, open that macro, copy the embedded code into the [Clipboard](#), then paste it into the expression to be used as the StyleText parameter.

#### StyleNote:

This character expression determines how the endnote number appears in the endnote itself. This information must be entered in the Style in Note box in the Endnote Options dialog box while recording a macro. This inserts an "embedded" code into the StyleNote character expression, which can be viewed using [Reveal Codes](#). If you want to insert an embedded code into an existing StyleNote parameter in a macro, record another macro containing that code, open that macro, copy the embedded code into the Clipboard, then paste it into the expression to be used as the StyleNote parameter.

#### LineSpacing:

This numeric expression specifies the line spacing within each endnote. This parameter is the equivalent of setting a value in the Line Spacing in Notes box in the Endnote Options dialog box.

**BetweenSpacing:**

This measurement expression specifies the amount of blank space that appears between each endnote. This parameter is the equivalent of setting a measurement in the Spacing Between Notes box in the Endnote Options dialog box.

**MinimumHeight:**

This measurement expression specifies the minimum amount of an endnote to keep together on a page. This parameter is the equivalent of setting a value in the Minimum Note Height box in the Endnote Options dialog box.

EndnoteOptionsDlg

---

## EndnoteOptionsDlg

The EndnoteOptionsDlg command displays the Endnote Options dialog box, which lets the user set numbering, spacing, and length options for endnotes. The options the user sets will apply from the insertion point in the active document forward. This command is the equivalent of choosing Endnote from the Layout menu, then choosing Options.

### **syntax**

EndnoteOptionsDlg ( )

EndnoteOptions

---

## EndnotePlacement

The EndnotePlacement command inserts a code so that all endnotes created before that point in the document, or from the point of the last Endnote Placement code, will be printed at that position in the document. This command also specifies whether to reset numbering for following endnotes. This command is the equivalent of choosing a button from the Endnote Placement dialog box.

### **syntax**

EndnotePlacement (Restart:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **Restart:**

This parameter specifies whether endnotes begin renumbering from the insertion point. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
No!	0	Endnotes do not begin renumbering from the insertion point.
Yes!	1	Endnotes begin renumbering from the insertion point.

EndnotePlacementDlg

---

## EndnotePlacementDlg

The EndnotePlacementDlg command displays the Endnote Placement dialog box, which lets the user specify whether to restart numbering endnotes at the insertion point. This command is the equivalent of choosing Endnote from the Layout menu, then choosing Placement.

### **syntax**

EndnotePlacementDlg ( )

EndnotePlacement

---

## EquationCaptionEdit

The EquationCaptionEdit command opens the Caption Editor for a specified equation, allowing following commands to edit the caption text for that equation. If an equation box is selected when this command is encountered, the Caption Editor opens for that box. If no equation box is selected when the EquationCaptionEdit command is encountered, the macro pauses and the Edit Equation Caption dialog box is displayed, which lets the user specify which equation box's caption to edit. Once the user specifies an equation, the Equation Caption Editor opens, and the macro resumes. The Caption Editor can be closed with the Close command.

This command is not recordable. To use the EquationCaptionEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing Equation from the Graphics menu, then choosing Caption, the macro facility records the BoxEditCaption ( ) command, which performs the same function.

### **syntax**

EquationCaptionEdit ( )

BoxEditCaption

Close

---

## EquationCreate

The EquationCreate command creates an equation box and opens the Equation Editor. The Equation Editor can be closed with the Close command. The EquationCreate command is the equivalent of choosing Equation from the Graphics menu, then choosing Create.

### **syntax**

EquationCreate ( )

Close

EquationCreateDlg

EquationEdit

---

## EquationCreateDlg

The EquationCreateDlg command displays the Select Editor dialog box, which lets the user specify whether the equation box being created will contain a figure, text, or an equation. When the user chooses OK, the appropriate editor opens, and the user can insert the contents of the box. The macro pauses until the user closes the editor, then resumes.

This command has no interface equivalent. This dialog box can only be invoked from a macro.

### **syntax**

EquationCreateDlg ( )

FigureCreateDlg

TextBoxCreateDlg



---

## EquationEdit

The EquationEdit command opens the Equation Editor. If an equation box is selected when this command is encountered, the Equation Editor opens with that box. If no equation box is selected when the EquationEdit command is encountered, the macro pauses and the Edit Equation dialog box is displayed, which lets the user specify which equation box to edit. Once the user specifies an equation to edit, that box is brought into the Equation Editor, and the macro resumes.

This command is not recordable. To use the EquationEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing Equation from the Graphics menu, then choosing Edit, the macro facility records the BoxEditor (Type:Equation!) command, which performs the same function.

### **syntax**

EquationEdit ( )

---

## EquationNewNumberDlg

The EquationNewNumberDlg command displays the Equation Number dialog box, which lets the user set a new starting number for equation boxes. This command is the equivalent of choosing Equation from the Graphics menu, then choosing New Number.

### **syntax**

EquationNewNumberDlg ( )

BoxNewNumber

---

## EquationOptionsDlg

The EquationOptionsDlg command displays the Equation Options dialog box, which lets the user set border, spacing, caption, and other options for equation boxes. This command is the equivalent of choosing Equation from the Graphics menu, then choosing Options.

### **syntax**

EquationOptionsDlg ( )

BoxOptions

---

## EquationPalette

Use this command to specify whether the Equation Palette in the Equation Editor should be displayed or hidden. EquationPalette is the equivalent of choosing Palette from the View menu in the Equation Editor.

### **syntax**

EquationPalette (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

If the palette is hidden when a macro encounters the command, the palette is displayed.

### **State:**

This parameter specifies whether the Equation Palette is displayed or hidden. The valid values for the State member set are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
Off!	0	The Equation Palette is hidden.
On!	1	The Equation Palette is displayed.

---

## EquationPositionDlg

The EquationPositionDlg command displays the Box Position and Size dialog box, which lets the user set the position, size, anchor, and other options for a specified equation box. This command can open the Position and Size dialog box in several ways:

If an equation box is being edited, the EquationPositionDlg command is the equivalent of choosing Box Position from the File menu in the Figure, Text, or Equation Editor.

If an equation box is currently selected when the EquationPositionDlg command is encountered, the Box Position and Size dialog box appears and lets the user set position and other options for that box.

If no box is selected when the EquationPositionDlg, the user can specify a number in the Equation Box Number text box indicating which box to modify. When the user has specified a number, the Box Position and Size dialog box is displayed, which lets the user set position and size options for that box.

### **syntax**

EquationPositionDlg ( )

BoxEditor

BoxSelect

EquationCreate

---

## EquationSettings

The EquationSettings command specifies the graphic font size, horizontal and vertical alignment of the equation, and whether the equation should be printed graphically for the current equation. The Equation Editor should be active whenever the EquationSettings command is encountered. This command is the equivalent of setting options in the Equation Settings dialog box.

### **syntax**

EquationSettings (*PrintAsGraphics:Value Set Member; FontSize:Numeric Expression; HorizontalAlignment:Value Set Member; VerticalAlignment:Value Set Member*)

### **parameters**

This command has four parameters:

#### **PrintAsGraphics:**

This parameter is the equivalent of choosing Print as Graphics to turn that option on or off. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Text!	0
Graphics!	1

#### **FontSize:**

This numeric expression sets the graphic font size for the equation. If the numeric expression is equal to 0, the default graphic font size is used. This parameter is the equivalent of typing a number in the Point Size text box.

#### **HorizontalAlignment:**

This parameter sets the horizontal alignment of the current equation in its box. This parameter is the equivalent of choosing an alignment from the Horizontal pop-up list. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Left!	0
Right!	1
Center!	2

#### **VerticalAlignment:**

This parameter sets the vertical alignment of the current equation in its box. This parameter is the equivalent of choosing an alignment from the Vertical pop-up list. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Top!	0
Center!	1
Bottom!	2

EquationEdit

EquationSettingsDlg

PrefEquation

---

## EquationSettingsDlg

The EquationSettingsDlg command displays the Equation Settings dialog box, which lets the user specify graphic font size, horizontal and vertical alignment of the equation, and whether the equation should be printed graphically. The Equation Editor should be active whenever the EquationSettingsDlg command is encountered. This command is the equivalent of choosing Settings from the File menu in the Equation Editor.

This dialog lets the user set the above parameters for individual equations. To set these parameters on a more global level, use PrefEquationDlg.

### **syntax**

EquationSettingsDlg ( )

EquationEdit

EquationSettings



---

## EquationZoom100

The EquationZoom100 command displays the equation in the Equation Editor viewing pane at 100%, so that the equation displays in the same size as it will be printed. This command is the equivalent of choosing 100% from the View menu in the Equation Editor.

### **syntax**

EquationZoom100 ( )

---

## EquationZoom200

The EquationZoom200 command resets the equation displayed in the Equation Editor viewing pane to its default size (200%). The EquationZoom200 command is the equivalent of choosing Zoom from the View menu, then choosing 200%.

### **syntax**

EquationZoom200 ( )

---

## EquationZoomFill

The EquationZoomFill command displays the equation in the Equation Editor viewing pane so that it fills the viewing pane. This command is the equivalent of choosing Zoom Fill from the View menu in the Equation Editor.

### **syntax**

EquationZoomFill ( )

---

## EquationZoomIn

The EquationZoomIn command displays the equation in the Equation Editor viewing pane at a larger percentage than it was previously displayed. This command is the equivalent of choosing Zoom In from the View menu in the Equation Editor.

### **syntax**

EquationZoomIn ( )

EquationZoomOut

---

## EquationZoomOut

The EquationZoomOut command displays the equation in the Equation Editor viewing pane at a smaller percentage than it was previously displayed. This command is the equivalent of choosing Zoom Out from the View menu in the Equation Editor.

### **syntax**

EquationZoomOut ( )

EquationZoomIn

---

## ExecuteFunction

The ExecuteFunction command executes a base function, one without parameters, without going through the third party chain. This command is usually used in conjunction with Macro Substitution so that even when a base function is unuseable through the normal methods because it has been substituted with a macro, you can bypass the third party chain and still use the function.

### **syntax**

ExecuteFunction (Function)

### **parameter**

This command has one parameter.

### **Function**

This parameter is a character expression. The Function is the name of the base function to execute.

**Example:** ExecuteFunction ("FileOpenDlg")

Macro Substitution

UserFunction

---

## FOR

FOR is the opening statement of the FOR-ENDFOR conditional loop. The FOR command begins a loop which executes all commands between the FOR and ENDFOR commands. The number of times this loop runs depends on the values given to the Initial Value, Test, and Increment parameters. For example, consider the command below:

```
For (VAR01; 3; VAR01<20; VAR01 + 2)
```

The first time through the loop, the variable in the Variable parameter (VAR01), is assigned the value of the Initial Value parameter (3). The macro then checks to see if the Test parameter (VAR01<20) is true. If the Test parameter is true, the commands between FOR and ENDFOR run. The Increment parameter (VAR01+2) then assigns the variable in the Variable parameter a new value. (In this case, since VAR01 had the value 3, its new value is 5.) The macro returns to the beginning of the FOR loop and runs, with the variable in the Variable parameter containing the new value. If the Test parameter is false, the macro goes to the command after ENDFOR and continues from that point. The steps remain the same every time through the loop, but the variable in the Variable parameter gets a new value from the Increment parameter each time the loop runs.

### **syntax**

FOR (*Variable; Initial Value; Test; Increment*)

### **parameters**

This command has four parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Initial Value**

This parameter is an alphanumeric, numeric, or measurement expression.

#### **Test**

Any relational expression is a valid value for this parameter.

#### **Increment**

This expression may be any expression, numeric or alphanumeric.

ENDFOR

FOREACH

Show Me!

---

## FOREACH

FOREACH is the opening statement of the FOREACH-ENDFOR conditional loop. The FOREACH command begins a loop that executes all commands between FOREACH and ENDFOR. The number of times this loop plays depends on the number of expressions specified between the braces ( { } ).

For instance, if the command has four Expression parameters, the loop runs four times. The first time the loop runs, the variable in the Variable parameter is assigned the value of the first Expression in the braces; the second time the loop runs, the variable is assigned the value of the second Expression in the braces, and so forth. When the loop has run with the variable containing the value of the final Expression in the braces, the macro goes to the command immediately below ENDFOR and continues.

### **syntax**

FOREACH (*Variable*; {*Expression*; *Expression*; . . . *Expression*})

### **parameters**

This command has two parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Expression**

Any numeric or character expression is a valid value for this parameter.

FOR

Show Me!



---

## FRACTION

The FRACTION command assigns the fractional portion of a numeric expression to a variable. For instance, to assign the fractional portion of the equation  $3/2$  (.5) to variable FRAC, you would type the following command:

```
Fraction(FRAC; 3/2)
```

Variable FRAC then contains the value 0.5.

### **syntax**

FRACTION(*Variable; Numeric Expression*)

### **parameters**

This command has two parameters:

#### **Variable:**

Any variable is valid for this parameter.

#### **Numeric Expression:**

Any numeric expression is valid for this parameter. Usually this numeric expression will contain some sort of mathematical operation, such as division.

## INTEGER

Show Me!

---

## FigureCaptionEdit

The FigureCaptionEdit command opens the Caption Editor for a specified figure, allowing following commands to edit the caption text for that figure. If a figure box is selected when this command is encountered, the Caption Editor opens for that box. If no figure box is selected when the FigureCaptionEdit command is encountered, the macro pauses and the Edit Figure Caption dialog box is displayed, which lets the user specify which figure box caption to edit. Once the user specifies a figure, the Figure Caption Editor opens, and the macro resumes. You can close the Caption Editor with the Close command.

This command is not recordable. To use the FigureCaptionEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing Figure from the Graphics menu, then choosing Caption, the macro facility records the BoxEditCaption ( ) command, which performs the same function.

### **syntax**

FigureCaptionEdit ( )

Close

BoxEditCaption

---

## FigureCreate

The FigureCreate command creates a figure box and opens the Figure Editor. The Figure Editor can be closed with the Close command. The FigureCreate command is the equivalent of choosing Figure from the Graphics menu, then choosing Create.

### **syntax**

FigureCreate ( )

Close

FigureEdit

FigureCreateDlg

---

## FigureCreateDlg

The FigureCreateDlg command displays the Select Editor dialog box, which lets the user specify whether the figure being created will contain a figure, text, or an equation. When the user chooses OK, the appropriate editor opens, and the user can insert the contents of the box. The macro pauses until the user closes the editor, then resumes. This command has no interface equivalent. This dialog box can be invoked only from a macro.

### **syntax**

FigureCreateDlg ( )

EquationCreateDlg

TextBoxCreateDlg

---

## FigureEdit

The FigureEdit command opens the Figure Editor. If a figure box is selected when this command is encountered, the Figure Editor opens with that box. If no figure box is selected when the FigureEdit command is encountered, the macro pauses and the Edit Figure dialog box is displayed, which lets the user specify which figure box to edit. Once the user specifies a figure to edit, that box is brought into the Figure Editor, and the macro resumes.

This command is not recordable. To use the FigureEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing Figure from the Graphics menu, then choosing Edit, the macro facility records the BoxEditor (Type:Figure!) command, which performs the same function.

### **syntax**

FigureEdit ( )

---

## FigureNewNumberDlg

The FigureNewNumberDlg command displays the Figure Number dialog box, which lets the user set a new number from which figure boxes can increment. This command is the equivalent of choosing Figure from the Graphics menu, then choosing New Number.

### **syntax**

FigureNewNumberDlg ( )

BoxNewNumber

---

## FigureOptionsDlg

The FigureOptionsDlg command displays the Figure Options dialog box, which lets the user set border, spacing, caption, and other options for figure boxes. This command is the equivalent of choosing Figure from the Graphics menu, then choosing Options.

### **syntax**

FigureOptionsDlg ( )

BoxOptions

---

## FigurePositionDlg

The FigurePositionDlg command displays the Box Position and Size dialog box, which lets the user set the position, size, anchor, and other options for a specified figure box. This command can open the Position and Size dialog box in several ways:

If a figure box is currently selected when the FigurePositionDlg command is encountered, the Position and Size dialog box is displayed, which lets the user set position and other options for that figure box.

If no box is selected, the user can specify a number in the Figure Box Number text box indicating which box to modify. When the user has specified a number, the Box Position and Size dialog box appears and lets the user set position and size options for that box.

If a figure box is currently being edited when this command is encountered, the command has no effect.

### **syntax**

FigurePositionDlg ( )

BoxEditor

BoxSelect

FigureCreate



---

## FigureRetrieveDlg

The FigureRetrieveDlg command displays the Retrieve Figure dialog box, which lets the user specify a figure to retrieve into the active document. This command is the equivalent of choosing Figure from the Graphics menu, then choosing Retrieve.

### **syntax**

FigureRetrieveDlg ( )

FigureCreate

---

## FileChangeDir

The FileChangeDir command changes either the Documents or Graphics Files directory to the path specified in the DosCurrent Dir parameter.

The specified directory remains active for the current session of WordPerfect. When you exit WordPerfect and restart it, each directory is reset to the path specified in the Location of Files dialog box.

### **syntax**

FileChangeDir (DosCurrentDir:*CharacterExpression*; WPDirectory:*Value Set Member*)

### **parameters:**

This command has two parameters

#### **DosCurrentDir**

This character expression specifies the directory to set as your current Documents directory or Graphics Files directory (depending on the value of the WPDirectory parameter). This command does not change the entries in the Location of Files dialog box.

#### **WPDirectory**

This value set specifies the WordPerfect directory that will be changed. The members and corresponding numeric equivalents of the WPDirectory value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

None!	0
GraphicsDir!	1
DocumentsDir!	2

---

## FileCopy

The FileCopy command copies a file specified in the parameter SourceFile to the file specified in the parameter DestinationFile. This command is the equivalent of selecting a file in the Open File dialog box, choosing Copy from the Options pop-up list, specifying a destination filename in the Copy File dialog box, then choosing Copy.

### **syntax**

FileCopy (SourceFile:*Character String*; DestinationFile:*Character String*)

### **parameters**

This command has two parameters:

#### **SourceFile:**

This character expression specifies the file to be copied. This is equivalent to typing a filename in the From text box of the Copy File dialog box.

#### **DestinationFile:**

This character expression specifies the file to which the source file should be copied. This is equivalent to typing a filename in the To text box of the Copy File dialog box. The user can enter a new filename to create a file or enter the name of an existing file, then choose OK to replace the file. A pathname can also be included.

FileDelete

FileMove

---

## FileDelete

The FileDelete command deletes the specified file. If no path is specified, WordPerfect looks in the current directory. This command is the equivalent of selecting a file in the Open File dialog box, choosing Delete from the Options pop-up list, then choosing Delete.

### **syntax**

FileDelete (Filename:*Character String*)

### **parameters**

This command has one parameter:

#### **Filename:**

This character expression specifies the file to be deleted. This is equivalent to typing a filename in the File to Delete text box of the Delete File dialog box. If no path is specified, this parameter causes WordPerfect to look for the file in the current DOS directory.

[FileCopy](#)

[FileMove](#)

---

## **FileFindDlg**

The FileFindDlg command brings up the File Find dialog box. This allows the user to search for a filename matching a specific pattern or to search for a file containing specific words. This command is the equivalent of choosing Find Files from File on the menu bar.

### **syntax**

FileFindDlg ( )

[FileOpenDlg](#)

---

## **FileManager**

The FileManager command opens the File Manager, which lets the user copy, delete, move, view, and otherwise manipulate files. This command is the equivalent of choosing File Manager from the File menu.

Once the File Manager is open, the macro is suspended until the user closes the File Manager. The macro then resumes playing.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

FileManager ( )

---

## FileMove

The FileMove command moves or renames a file specified in the parameter SourceFile to the file specified in the parameter DestinationFile. This command is the equivalent of selecting a file in the Open File dialog box, choosing Move/Rename from the Options pop-up list, specifying a destination filename in the Move/Rename File dialog box, then choosing Move.

### **syntax**

FileMove (SourceFile:*Character String*; DestinationFile:*Character String*)

### **parameters**

This command has two parameters:

#### **SourceFile:**

This character expression specifies the file to be moved or renamed. This is equivalent to typing a filename in the From text box of the Move/Rename File dialog box.

#### **DestinationFile:**

This character expression specifies the file to which the source file should be moved or renamed. This is equivalent to typing a filename in the To text box of the Move/Rename File dialog box. A pathname can also be included.

[FileDelete](#)

[FileCopy](#)

---

## **FileNew**

The FileNew command opens a new, blank document, providing that fewer than nine documents are currently open. If nine documents are already open, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. This command is the equivalent of choosing New from the File menu.

### **syntax**

FileNew ( )

LABEL

ONERROR



---

## FileOpen

The FileOpen command retrieves a specified document file into a new document window. This command is the equivalent of specifying a document to open in the Open File dialog box, then choosing Open. If 9 document windows are currently open, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command.

### **syntax**

FileOpen (Filename:*Character Expression*; AutoDetect:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **Filename:**

This character expression specifies the name of the document file to be opened. This parameter is the equivalent of typing a filename in the Filename box in the Open File dialog box.

#### **AutoDetect:**

This parameter specifies whether or not WordPerfect suppresses the Convert File Format dialog box when a non-WordPerfect 5.1/5.2 format file is being opened. The members and corresponding numeric equivalents of the AutoDetect value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

LABEL

ONERROR

Close

FileOpenDlg

LastOpenedAdd

FileFindDlg

---

## **FileOpenDlg**

The FileOpenDlg command displays the Open File dialog box, which lets the user specify a file to be opened into a new document window. This command is the equivalent of choosing Open from the File menu.

### **syntax**

FileOpenDlg ( )

FileOpen

---

## FileRetrieve

The FileRetrieve command retrieves a specified file into the active document at the insertion point. This command is the equivalent of specifying a file to retrieve in the Retrieve File dialog box, then choosing Retrieve.

### syntax

FileRetrieve (Filename:*Character Expression*; AutoDetect:*Value Set Member*; InsertIntoDoc:*Value Set Member*)

### parameters

This command has three parameters:

#### Filename:

This character expression specifies just the filename or the entire pathname of the file to be retrieved into the active document window.

#### AutoDetect:

This parameter specifies whether WordPerfect displays the Convert File Format dialog box when a non-WordPerfect 5.1 format file is being opened. The members and corresponding numeric equivalents of the AutoDetect value set are:

Member	Numeric Equivalent	Definition
No!	0	The dialog box is displayed.
Yes!	1	The dialog box is not displayed, but the file is converted.

#### InsertIntoDoc:

This parameter specifies whether WordPerfect displays the "Insert file into current document?" message box that appears when a file would be inserted into the active document. The members and corresponding numeric equivalents of the InsertIntoDoc value set are:

Member	Numeric Equivalent	Definition
Prompt!	0	The message box is not suppressed.
Insert!	1	The message box is suppressed, but the file is inserted.

FileRetrieveDlg

---

## FileRetrieveDlg

The FileRetrieveDlg command displays the Retrieve File dialog box, which lets the user specify a file to retrieve into the active document. The file is retrieved at the insertion point. This command is the equivalent of choosing Retrieve from the File menu.

### **syntax**

FileRetrieveDlg ( )

FileRetrieve

---

## FileSave

The FileSave command saves the active document. If the document has never been saved before, the Save As dialog box appears and the macro pauses while the user types the name of the document. Once the user has entered a document name, the macro continues. If the document already has a name, the FileSave command saves it by that name.

This command is the equivalent of choosing Save from the File menu. The default directory for files saved using this command is the current default directory.

### **syntax**

FileSave (Filename:*Character Expression*; ExportType:*Value Set Member*; Overwrite:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **Filename:**

This character expression is the equivalent of typing a filename in the Save As text box in the Save As dialog box.

#### **ExportType:**

This value set member is the equivalent of choosing a file format from the Format pop-up list. The members and corresponding numeric equivalents of the ExportType value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
WordPerfect42!	1
WordPerfect50!	2
WordPerfect51!	3
WordStar33!	30
WordStar331!	31
WordStar34!	32
WordStar40!	33
WordStar50!	34
WordStar55!	35
WordStar60!	36
MicrosoftWord50!	63
MicrosoftWord55!	64
WordForWindows10!	70
WordForWindows11!	71
IBMDCARevisableFormText!	80
IBMDCAFinalFormText!	81
DisplayWrite40!	82
DisplayWrite42!	83
DisplayWrite50!	84
MultiMate33!	100

MultiMateAdvantage36!	101
MultiMateAdvantageII37!	102
MultiMate40!	103
OfficeWriter60!	120
OfficeWriter61!	121
OfficeWriter611!	122
OfficeWriter62!	123
WordRichTextFormat!	150
XyWriteIIIPlus355!	190
XyWriteIIIPlus356!	191
AmiPro12Windows!	311
AmiPro12aWindows!	312
AmiPro12bWindows!	313
ANSIDelimitedTextWindows!	487
StrippedANSIWindows!	488

**Overwrite:**

This value set member allows the user to specify if the file should be overwritten or if the user should be prompted if the file already exists. If the value is set to No! and a file with the same name exists, an error is generated and macro execution will be terminated or redirected to the LABEL specified by ONERROR. If the value is set to Yes!, the file will be overwritten without prompting the user. If the value is set to Prompt!, and a file with the same name already exists, the Save As dialog box will be displayed and the user will be prompted to type a unique filename. The members and corresponding numeric equivalents of the Overwrite value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
No!	0
Yes!	1
Prompt!	2

LABEL

ONERROR

FileSaveAsDlg

---

## FileSaveAsDlg

The FileSaveAsDlg command displays the Save As dialog box, which lets the user specify a name for the active document. This command is the equivalent of choosing Save As from the File menu.

### **syntax**

FileSaveAsDlg ( )

FileSave

---

## Font

The Font command lets you specify the font used for your text. The font you choose is in effect from the insertion point forward, until another font code is encountered. This command is the equivalent of setting a typeface and size in the Font dialog box.

### **syntax**

Font (Name:*Character Expression*; Size:*Measurement Expression*; ForceInsert:*Value Set Member*)

### **parameters**

This command has three parameters.

#### **Name:**

This character expression specifies the typeface used. This parameter should correspond to a font available on the selected printer.

#### **Size:**

This measurement expression specifies the height of the font in points (72 points equal one inch). This parameter should only be used with printer drivers that have scalable fonts available. If the currently selected printer driver does not have scalable fonts, this parameter should be left blank.

#### **ForceInsert:**

This parameter specifies whether to insert the font even if it is the same as the current font. Normally, WordPerfect does not insert a font code unless it differs from the current font. A Yes! value forces the insertion of the font code into the document without checking to see if it is the same as the current font.

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

FontDlg



---

## FontBold

Use FontBold to turn the Bold attribute on or off at the insertion point, or to bold selected text. This command is the equivalent of choosing Bold from the Font menu.

### **syntax**

FontBold (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Bold is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose bold from the Font menu, the State parameter for the FontBold command is set to On!.

FontNormal

---

## FontColor

The FontColor command inserts a Color code into the active document at the insertion point or changes the color of selected text. The Color code changes the color of text on the screen, as well as the print color, if the printer being used has color capability. The FontColor command is the equivalent of setting a custom color in the Select Text Color dialog box.

### **syntax**

FontColor (AmountRed:*Numeric Expression*; AmountGreen:*Numeric Expression*;  
AmountBlue:*Numeric Expression*)

### **parameters**

This command has three parameters:

#### **AmountRed: AmountGreen: AmountBlue:**

These three numeric expressions correspond to the Red, Green, and Blue boxes in the Color Options group in the Select Text Color dialog box. These three parameters should each be integers ranging from 0 to 255. The combination of these three parameters determines the actual color to be displayed and printed.

FontColorDlg

---

## FontColorDlg

The FontColorDlg command displays the Select Text Color dialog box, which lets the user insert a Color code into the active document at the insertion point. The Color code changes the color of text on the screen, as well as the print color, if the printer being used has color capability. This command is the equivalent of choosing Color from the Font menu.

### **syntax**

FontColorDlg ( )

FontColor

---

## FontDlg

The FontDlg command displays the Font dialog box, which lets the user select the font to be used from the insertion point forward. This command is the equivalent of choosing Font from the Font menu.

### **syntax**

FontDlg ( )

Font

---

## FontDoubleUnderline

Use this command to turn the Double Underline attribute on or off at the insertion point, or to double underline selected text. This command is the equivalent of choosing Double Underline from the Font menu.

### **syntax**

FontDoubleUnderline (State:Value Set Member)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Double Underline is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose DoubleUnderline from the Font menu, the State parameter for the FontDoubleUnderline command is set to On!.

FontNormal

---

## FontExtraLarge

Use this command to turn the Extra Large attribute on or off at the insertion point, or to make selected text Extra Large. This command is the equivalent of choosing Size from the Font menu, then choosing Extra Large.

### **syntax**

FontExtraLarge (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether the attribute is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose ExtraLarge from Size on the Font menu, the State parameter for the FontExtraLarge command is set to On!.

FontNormal

---

## FontFine

Use this command to turn the Fine attribute on or off at the insertion point, or to change the size of selected text to Fine. This command is the equivalent of choosing Size from the Font menu, then choosing Fine.

### **syntax**

FontFine (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether the attribute is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Fine from Size on the Font menu, the State parameter for the FontFine command is set to On!.

### FontNormal

---

## FontItalic

Use this command to turn the Italic attribute on or off at the insertion point, or to italicize selected text. This command is the equivalent of choosing Italic from the Font menu.

### **syntax**

FontItalic (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Italics is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Italic from the Font menu, the State parameter for the FontItalic command is set to On!.

FontNormal



---

## FontLarge

Use this command to turn the Large attribute on or off at the insertion point, or to change the size of selected text to Large. This command is the equivalent of choosing Size from the Font menu, then choosing Large.

### **syntax**

FontLarge (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether the attribute is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Large from Size on the Font menu, the State parameter for the FontLarge command is set to On!.

### FontNormal

---

## FontNormal

The FontNormal command turns off all active font attributes (such as bold, italic, and underline). This command is the equivalent of choosing Normal from the Font menu.

### **syntax**

FontNormal ( )

---

## FontOutline

Use the FontOutline command to turn the Outline font attribute on or off at the insertion point, or to outline selected text. This command is the equivalent of selecting Outline in the Appearance group box in the Font dialog box.

When the insertion point is located in text already surrounded by [Outln On] and [Outln Off] codes and the macro encounters a FontOutline (State:Off!) command, the Outline attribute is turned off for the insertion point, leaving the text on both sides outlined. You will see [Outln On] and [Outln Off] codes surrounding the text before and after the insertion point, indicating that text typed at the insertion point will not be outlined, but the text surrounding it will remain outlined.

If, while recording a macro, you select non-outlined text and select Outline in the Appearance group box in the Fonts dialog box, a FontOutline (State:On!) command is recorded in the macro.

### **syntax**

FontOutline (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

#### **State:**

This parameter determines whether the Outline appearance attribute is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

FontNormal

---

## FontRedline

Use this command to turn the Redline attribute on or off at the insertion point, or to redline selected text. This command is the equivalent of choosing Redline from the Font menu.

### **syntax**

FontRedline (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Redline is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Redline from the Font menu, the State parameter for the FontRedline command is set to On!.

FontNormal

---

## FontShadow

Use this command to turn the Shadow attribute on or off at the insertion point, or to shadow selected text. This command is the equivalent of choosing Shadow from the Appearance group box in the Font dialog box.

### **syntax**

FontShadow (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Shadow is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Shadow from the Font dialog box on the Font menu, the State parameter for the FontShadow command is set to On!.

### FontNormal

---

## FontSmall

Use this command to turn the Small attribute on or off at the insertion point, or to change the size of selected text to Small. This command is the equivalent of choosing Size from the Font menu, then choosing Small.

### **syntax**

FontSmall (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether the attribute is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Small from Size on the Font menu, the State parameter for the FontSmall command is set to On!.

### FontNormal

---

## FontSmallCaps

Use this command to turn the Small Caps attribute on or off at the insertion point, or to change selected text to small capital letters. This command is the equivalent of choosing Small Cap from the Appearance group in the Font dialog box.

### **syntax**

FontSmallCaps (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Small Caps is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Small Cap from the Font dialog box on the Font menu, the State parameter for the FontSmallCaps command is set to On!.

FontNormal

---

## FontStrikeout

Use this command to turn the Strikeout attribute on or off at the insertion point, or to strike out selected text. This command is the equivalent of choosing Strikeout from the Font menu.

### **syntax**

FontStrikeout (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Strikeout is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Strikeout from the Font menu, the State parameter for the FontStrikeout command is set to On!.

FontNormal



---

## FontSubscript

Use this command to turn the Subscript attribute on or off at the insertion point, or to subscript selected text. This command is the equivalent of choosing Subscript from the Font menu.

### **syntax**

FontSubscript (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Subscript is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Subscript from the Font menu, the State parameter for the FontSubscript command is set to On!.

FontNormal

---

## FontSuperscript

Use this command to turn the Superscript attribute on or off at the insertion point, or to superscript selected text. This command is the equivalent of choosing Superscript from the Font menu.

### **syntax**

FontSuperscript (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Superscript is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Superscript from the Font menu, the State parameter for the FontSuperscript command is set to On!.

FontNormal

---

## FontUnderline

Use this command to turn the Underline attribute on or off at the insertion point, or to underline selected text. This command is the equivalent of choosing Underline from the Font menu.

### **syntax**

FontUnderline (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Underline is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Underline from the Font menu, the State parameter for the FontUnderline command is set to On!.

FontNormal

---

## FontVeryLarge

Use this command to turn the Very Large attribute on or off at the insertion point, or to change the size of selected text to Very Large. This command is the equivalent of choosing Size from the Font menu, then choosing Very Large.

### **syntax**

FontVeryLarge (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter determines whether Very Large is turned on or off. The members and corresponding numeric equivalents of the State value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

If, while recording a macro, you select text and choose Very Large from Size on the Font menu, the State parameter for the FontVeryLarge command is set to On!.

### FontNormal

---

## FooterDlg

The FooterDlg command displays the Footers dialog box, which lets the user specify whether to create, edit, or discontinue a specified footer. This command is the equivalent of choosing Page from the Layout menu, then choosing Footers.

### **syntax**

FooterDlg ( )

HeaderFooter

---

## FootnoteCreate

The FootnoteCreate command opens a Footnote window. Following commands can then insert text and codes into this window. When the footnote is completed, a Close command can close the Footnote window, placing a reference to the footnote at the insertion point in the active document. The FootnoteCreate command is the equivalent of choosing Footnote from the Layout menu, then choosing Create.

### **syntax**

FootnoteCreate ( )

Close

---

## FootnoteEdit

The FootnoteEdit command opens the Footnote window for a specified footnote. The Footnote window contains the text and codes for the footnote specified in the FootnoteNumber parameter. Once this window is open, all macro commands apply to the Footnote window until the Close command is encountered. The FootnoteEdit command is the equivalent of choosing Footnote from the Layout menu, then choosing Edit.

### **syntax**

FootnoteEdit (FootnoteNumber:*Character Expression*)

### **parameters**

This command has one parameter. If no parameter is specified with this command, the footnote following the insertion point is edited.

### **FootnoteNumber:**

This character expression specifies the number of the footnote to be edited. Note that this parameter is a character expression, not a numeric expression. This is because footnote numbers are not always expressed as arabic numerals; they may also be expressed as roman numerals or letters.

Close

FootnoteEditDlg

---

## FootnoteEditDlg

The FootnoteEditDlg command displays the Edit Footnote dialog box, which lets the user specify which footnote to edit in the active document. This command is the equivalent of choosing Footnote from the Layout menu, then choosing Edit.

### **syntax**

FootnoteEditDlg ( )

FootnoteEdit



---

## FootnoteNewNumber

The FootnoteNewNumber command inserts a code at the insertion point, so that following footnotes will begin incrementing from the number specified in this command. This command is the equivalent of entering a number in the Footnote Number dialog box.

### **syntax**

FootnoteNewNumber (FootnoteNumber:*Character Expression*)

### **parameters**

This command has one parameter:

#### **FootnoteNumber:**

This character expression specifies the number from which following footnotes will be incremented. Note that this parameter is a character expression, not a numeric expression. This is because Footnote numbers are not always expressed as arabic numerals; they may also be expressed as roman numerals or letters.

FootnoteNewNumberDlg

---

## FootnoteNewNumberDlg

The FootnoteNewNumberDlg command displays the Footnote Number dialog box, which lets the user set a number from which following footnotes will be incremented. This command is the equivalent of choosing Footnote from the Layout menu, then choosing New Number.

### **syntax**

FootnoteNewNumberDlg ( )

FootnoteNewNumber

---

## FootnoteOptions

The FootnoteOptions command sets options for all footnotes following the [insertion point](#). These options include numbering options, printing options, positioning options, amount of space between notes, and the length a note must be before it can span a page break. The FootnoteOptions command is the equivalent of setting options in the Footnote Options [dialog box](#).

### syntax

FootnoteOptions (NumberingMethod:*Value Set Member*; Characters:*Character Expression*; StyleText:*Character Expression*; StyleNote:*Character Expression*; LineSpacing:*Measurement Expression*; BetweenSpacing:*Measurement Expression*; MinimumHeight:*Measurement Expression*; Restart:*Value Set Member*; PrintContinued:*Value Set Member*; FootnotePosition:*Value Set Member*; Separator:*Value Set Member*)

### parameters

This command has 11 parameters:

#### NumberingMethod:

This parameter specifies whether numbers, letters, or characters will be used as the footnote markers. The NumberingMethod value set has three valid values, corresponding to the items in the Numbering Method pop-up list in the Footnote Options dialog box:

Member	Numeric Equivalent
Numbers!	0
Characters!	1
Letters!	2

#### Characters:

This parameter is only necessary when the Style parameter of this command is set to Characters!. The Characters parameter specifies which character(s) to use in footnote markers. This character expression should be no more than four characters in length.

#### StyleText:

This character expression determines how the footnote number appears in the text of the document. For example, this parameter determines whether the footnote number will be italicized and superscript or bolded and subscript. This information must be entered in the Style in Text box while recording a macro. This inserts an "embedded" code into the StyleText character expression, which can be viewed using [Reveal Codes](#). If you want to insert an embedded code into an existing StyleText parameter in a macro, record another macro containing that code, open that macro, copy the embedded code in the [Clipboard](#), then paste it into the expression to be used as the StyleText parameter.

#### StyleNote:

This character expression determines how the footnote number appears in the footnote itself. For example, whether the footnote number will be italicized and superscript or bolded and subscript. This information must be entered in the Style in Note box while recording a macro. This inserts an "embedded" code into the StyleNote character expression, which can be viewed using [Reveal Codes](#). If you want to insert an embedded code into an existing StyleNote parameter in a macro, record another macro containing that code, open that macro, copy the embedded code in the [Clipboard](#), then paste it into the expression to be used as the StyleNote parameter.

#### LineSpacing:

This numeric expression specifies the line spacing within each footnote. This parameter is the equivalent of setting a value in the Line Spacing in Notes box in the Footnote Options dialog box.

**BetweenSpacing:**

This measurement expression specifies the amount of blank space that appears between each footnote. This parameter is the equivalent of setting a measurement in the Spacing Between Notes box in the Footnote Options dialog box.

**MinimumHeight:**

This measurement expression specifies the minimum amount of a footnote to keep together on a page. This parameter is the equivalent of setting a value in the Minimum Note Height box in the Footnote Options dialog box.

**Restart:**

This parameter specifies whether footnote numbering will begin at 1 on every page, corresponding to the Restart Numbering on Each Page check box in the Footnote Options dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Description</b>
Off!	0	Do not restart numbering on each page.
On!	1	Restart numbering on each page.

**PrintContinued:**

This parameter specifies whether "(continued)..." is printed on the last footnote line of the first page and "...(continued)" is printed on the first footnote line of the next page, corresponding to the Print (Continued . . . ) Message check box in the Footnote Options dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Description</b>
No!	0	Do not print "continued" messages.
Yes!	1	Print "continued" messages.

**FootnotePosition:**

This parameter specifies whether footnotes will be printed at the bottom of the page or immediately below the last line of text. This parameter is the equivalent of choosing an option from the Position pop-up list in the Footnote Options dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
AfterText!	0
BottomOfPage!	1

**Separator:**

This parameter specifies whether no line, a short line, or a line spanning from the left to the right margins will separate the document text from the footnotes. This parameter is the equivalent of selecting an option from the Separator pop-up list in the Footnote Options dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
NoLine!	0
ShortLine!	1

MarginToMargin! 2

FootnoteOptionsDlg

---

## FootnoteOptionsDlg

The FootnoteOptionsDlg command displays the Footnote Options dialog box, which lets the user set numbering, spacing, and length options for footnotes. The options set will apply from the insertion point forward in the active document. This command is the equivalent of choosing Footnote from the Layout menu, then choosing Options.

### **syntax**

FootnoteOptionsDlg ( )

FootnoteOptions

---

## **ForcePageEven**

Use this command to make certain that the page containing the insertion point will be an even-numbered page. If necessary, WordPerfect will insert a blank page in front of the current page to ensure that the current page is even. This command is the equivalent of selecting the Even check box in the Page Numbering dialog box.

### **syntax**

ForcePageEven ( )

ForcePageOdd

PageNumberingDlg

---

## **ForcePageOdd**

Use this command to make certain that the page containing the insertion point will be an odd-numbered page. If necessary, WordPerfect will insert a blank page in front of the current page to ensure that the number of the current page is odd. This command is the equivalent of selecting the Odd check box in the Page Numbering dialog box.

### **syntax**

ForcePageOdd ( )

ForcePageEven

PageNumberingDlg



---

## GETNUMBER

The GETNUMBER command displays a dialog box with the prompt specified in the Prompt parameter and the title specified in the Title parameter. The macro pauses while the user types a number.

After the user presses Enter or chooses OK, WordPerfect checks the user's entry to make sure it is a number. If it is not a number, a message box appears and tells the user to re-enter the number. When the user chooses OK in this message box, the GETNUMBER dialog box reappears and the user must type a new number. When the user enters an acceptable number, the variable in the Variable parameter is assigned that number. If the user chooses Cancel, the dialog box closes and the macro acts as if a cancel has been encountered.

When the macro plays, the command appears as a dialog box with the message and title you specify.

### **syntax**

GETNUMBER (*Variable; Prompt; Title*)

### **parameters**

This command has three parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Prompt**

This character expression instructs the user to enter a certain type of number.

#### **Title**

This character expression appears in the title bar of the dialog box.

GETSTRING

GETUNITS

NUMSTR

ONCANCEL

Show Me!

---

## GETSTRING

The GETSTRING command displays a dialog box with the prompt specified in the Prompt parameter and the title specified in the Title parameter. The macro pauses while the user types. If the optional LENGTH = Numeric Expression parameter is included in this command, the dialog box will not allow the user to type in more text than the maximum number of bytes specified. After the user presses Enter or chooses OK, the variable in the Variable parameter is assigned that text. If the user chooses Cancel, the dialog box closes and the macro acts as if a cancel has been encountered.

When the macro plays, the command appears as a dialog box with the message and title you specify.

### **syntax**

GETSTRING (*Variable; LENGTH = Numeric Expression; Prompt; Title*)

### **parameters**

This command has four parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **LENGTH = Numeric Expression**

This numeric expression specifies in bytes the maximum allowable length of text to be entered in this dialog box. Standard characters are each one byte long. However, extended characters may take more than one byte each. The numeric expression should be preceded by the text "LENGTH =".

#### **Prompt**

This character expression instructs the user to type a certain type of information.

#### **Title**

This character expression appears in the title bar of the dialog box.

GETNUMBER

GETUNITS

STRNUM

Show Me!

---

## GETUNITS

The GETUNITS command displays a dialog box with the prompt specified in the Prompt parameter and the title specified in the Title parameter. The macro pauses while the user types a measurement. The measurement may include a unit of measurement. If no unit of measurement is specified, the macro uses 1/1200 of an inch, or the unit of measurement specified by the DEFAULTUNITS command.

After the user presses Enter or chooses OK, WordPerfect checks to make sure the number is a valid measurement. If it is not, a message box appears, which tells the user to re-type the measurement. When the user chooses OK in this message box, the GETUNITS dialog box reappears and a new measurement must be typed. When the user enters an acceptable measurement, the variable in the Variable parameter is assigned that measurement.

If the user chooses Cancel, the dialog box closes and the macro acts as if a cancel has been encountered.

When the macro plays, the command appears as a dialog box with the message and title you specify.

### **syntax**

GETUNITS (*Variable; Prompt; Title*)

### **parameters**

This command has three parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Prompt**

This character expression instructs the user to enter a certain type of measurement expression.

#### **Title**

This character expression appears in the title bar of the dialog box.

DEFAULTUNITS

GETSTRING

GETNUMBER

Show Me!

---

## **GO**

The GO command causes macro execution to jump to the LABEL command specified in the Label parameter, then to continue from there.

### **syntax**

Go (*Label@*)

### **parameters**

This command has one parameter:

#### **Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

LABEL

Show Me!

---

## Generate

The generate command generates lists, indexes, tables of contents, and tables of authorities in the active document, replacing any existing generated text. This command is the equivalent of choosing Generate from the Tools menu, then choosing Yes.

### **syntax**

Generate ( )

GenerateDlg

---

## GenerateDlg

The GenerateDlg command displays the Generate dialog box, which gives the user the option to generate the document addenda (for example, lists, indexes, and tables of contents) in the active document window. This command is the equivalent of choosing Generate from the Tools menu.

### **syntax**

GenerateDlg ( )

Generate

---

## GetWPData

Use the GetWPData command to assign specific information about the state of WordPerfect or WordPerfect documents to a variable. The macro can then use that variable in conjunction with other commands to decide whether to perform certain actions. For example, you may want to open a new document, but only if the active document isn't new. You may want to move the insertion point to page 17, but only if it isn't already there. You may even want to determine the character to the left or right of the insertion point.

The type of information assigned to the variable specified in the MacroVariable parameter depends on the system variable you specify in the SystemVariable parameter. The value type will be a character expression, a numeric expression, or a boolean value.

### syntax

GetWPData (MacroVariable:Variable; SystemVariable:Value Set Member)

### parameters

This command has two parameters:

#### MacroVariable:

The information contained in the system variable specified in the SystemVariable parameter is assigned to this variable. Any variable name is a valid value for this parameter.

#### SystemVariable:

This parameter specifies which system variable to check. The value of the specified system variable is assigned to the variable specified in the MacroVariable parameter. The members of this value set are listed in the table below. You can use the member name or its numeric equivalent in the command syntax.

Member	Numeric Equivalent	Description
Cell!	2	Current cell position in a table (for example, A4 or E7). This system variable is A1 if the insertion point is not in a table when the command is encountered. The cell position is assigned to the variable as a character expression.
Column!	3	Current column number (numbered sequentially from left to right) in a table or in columns. The column number is assigned to the variable as a numeric expression.
Endnote!	5	Number of the current endnote, assigned to the variable as a numeric expression.
Equation!	6	Number of the current equation, according to the following formula: Return value/32=first level; Return value %32=second level. For example, if the current equation is 1.2 and the macro encounters GetWPData (MacroVariable:EquNumber; SystemVariable:Equation!), the

EquNumber variable will contain the numeric value 34. (In other words,  $34/32=1$ ,  $34\%32=2$ .)

Figure!	7	Number of the current figure, according to the formula described under system variable Equation above.
Footnote!	8	Number of the current footnote.
LeftCode!	9	The numeric equivalent of the code immediately to the left of the insertion point. If a character, instead of a code, is immediately to the left of the insertion point, the variable will be assigned a value of 0. You can then use the LeftChar! value set member to locate the character to the left of the insertion point. (For a list of the numeric equivalents for WordPerfect codes, refer to WordPerfect Codes below.)
Line!	10	Vertical position of the insertion point in WordPerfect units (i.e., 1200ths of an inch). A measurement value is assigned to the variable.
Name!	12	Name of the current document (for example, JONES.LTR,). This value is assigned to the variable as a character expression. This value set member returns an empty string when played from an unnamed document.
Page!	14	Current page number. This value is assigned to the variable as a numeric expression.
Path!	15	Path to the current document (for example, C:\WPWIN\). (Note the backslash on the end of the path.) This member is assigned to the variable as a character expression. This value set member returns an empty string when played from an unnamed document.
Pos!	16	Current cursor position in WordPerfect units (i.e., 1200ths of an inch). A measurement value is assigned to the variable.
RightCode!	18	The numeric equivalent of the code immediately to the right of the insertion point. If a character, instead of a code, is immediately to the right of the insertion



point, the variable will be assigned a value of 0. You can then use the RightChar! value set member to locate the character to the right of the insertion point. (For a list of the numeric equivalents for WordPerfect codes, refer to WordPerfect Codes below.)

TableBox!	19	Number of the current table box or table, according to the formula described under system variable Equation! above.
TextBox!	20	Number of the current text box, according to the formula described under system variable Equation! above.
UserBox!	21	Number of the current user box, according to the formula described under system variable Equation! above.
Row!	22	Current row number in a table (equals 0 if the cursor is not in a table when the command is encountered).
RowState!	27	Header status of current row in a table. Variable is assigned a boolean value, with TRUE meaning the current row is a header row, and FALSE meaning either the current row is not a header row or the insertion point is not in a table.
Network!	30	The variable is assigned a boolean value specifying whether the network version of WordPerfect is currently running. TRUE means WordPerfect is running on a network; FALSE means WordPerfect is not running on a network.
Language!	32	Current language in use. The variable is assigned a two-character character expression representing the name of the active language module. For example, American English is US, British English is UK, and so forth.
FontNormal!	64	The variable is assigned a boolean value specifying whether any size or appearance attributes are on at the insertion point. TRUE means no attributes are on; FALSE means one or more size or appearance attributes are on at the insertion point.
FontExtraLarge!	65	The variable is assigned a boolean value specifying whether this attribute is on at the insertion point. TRUE means the

attribute is on; FALSE means the attribute is off.

FontVeryLarge!	66	See FontExtraLarge! above.
FontLarge!	67	See FontExtraLarge! above.
FontSmall!	68	See FontExtraLarge! above.
FontFine!	69	See FontExtraLarge! above.
FontSuperscript!	70	See FontExtraLarge! above.
FontSubscript!	71	See FontExtraLarge! above.
FontOutline!	72	See FontExtraLarge! above.
FontItalics!	73	See FontExtraLarge! above.
FontShadow!	74	See FontExtraLarge! above.
FontRedline!	75	See FontExtraLarge! above.
FontDoubleUnderline!	76	See FontExtraLarge! above.
FontBold!	77	See FontExtraLarge! above.
FontStrikeout!	78	See FontExtraLarge! above.
FontUnderline!	79	See FontExtraLarge! above.
FontSmallCaps!	80	See FontExtraLarge! above.
CellNormal!	81	The variable is assigned a boolean value specifying whether any size or appearance attributes are on in the current cell. TRUE means no attributes are on; FALSE means one or more size or appearance attributes are on in the current cell.
CellExtraLarge!	82	The variable is assigned a boolean value specifying whether this attribute is on in the current cell. TRUE means the attribute is on; FALSE means the attribute is off.
CellVeryLarge!	83	See CellExtraLarge! above.
CellLarge!	84	See CellExtraLarge! above.
CellSmall!	85	See CellExtraLarge! above.
CellFine!	86	See CellExtraLarge! above.
CellSuperscript!	87	See CellExtraLarge! above.

CellSubscript!	88	See CellExtraLarge! above.
CellOutline!	89	See CellExtraLarge! above.
CellItalics!	90	See CellExtraLarge! above.
CellShadow!	91	See CellExtraLarge! above.
CellRedline!	92	See CellExtraLarge! above.
CellDoubleUnderline!	93	See CellExtraLarge! above.
CellBold!	94	See CellExtraLarge! above.
CellStrikeout!	95	See CellExtraLarge! above.
CellUnderline!	96	See CellExtraLarge! above.
CellSmallCaps!	97	See CellExtraLarge! above.
DocumentModified!	98	The variable is assigned a boolean value specifying whether the active document has been modified. TRUE means the document has been modified; FALSE means the document has not been modified.
DocumentNeedsGenerating!	99	The variable is assigned a boolean value specifying whether the active document has been modified since it was last generated. TRUE means the document has been modified since it was last generated; FALSE means the document has not been modified since it was last generated.
DocumentBlank!	100	The variable is assigned a boolean value specifying whether the active document is blank. A blank document is a document that has never contained any codes or text. A document that has had all codes and text erased is not considered blank. TRUE means the document is blank; FALSE means the document is not blank.
BetweenTableCodes!	101	The variable is assigned a boolean value specifying whether the insertion point is between [Tbl Def] and [Tbl Off] codes. TRUE means the insertion is between table codes; FALSE means the insertion point is not between table codes.
BetweenMathCodes!	102	The variable is assigned a boolean value specifying whether the insertion point is

		between [Math On] and [Math Off] codes. TRUE means the insertion is between math codes; FALSE means the insertion point is not between math codes.
BetweenOutlineCodes!	103	The variable is assigned a boolean value specifying whether the insertion point is between [Outline On] and [OutlineOff] codes. TRUE means the insertion is between outline codes; FALSE means the insertion point is not between outline codes.
CellJustification!	104	The variable is assigned an integer from 0 to 4, with 0 meaning the cell is left-justified, 1 meaning the cell is right-justified, 2 meaning the cell is full-justified, 3 meaning the cell is center-justified, and 4 meaning the cell is decimal-aligned.
JustifyCellSpecific!	105	The variable is assigned a boolean value specifying whether the justification of the cell containing the insertion point is applied to that specific cell, rather than to the entire table or document. TRUE means justify is cell-specific, FALSE means justify is not cell-specific.
AttributeCellSpecific!	106	The variable is assigned a boolean value specifying whether any attributes in the cell containing the insertion point are applied to that specific cell, rather than to the entire table or document. TRUE means at least one attribute is cell-specific; FALSE means that attributes are not cell-specific.
CellBottomAligned!	107	The variable is assigned a boolean value specifying whether the current cell is bottom aligned. TRUE means the cell is bottom-aligned; FALSE means the cell is not bottom-aligned.
CellCenterAligned!	108	The variable is assigned a boolean value specifying whether the current cell is center aligned. TRUE means the cell is center-aligned; FALSE means the cell is not center-aligned.
CellIgnoredWhileCalculating!	109	The variable is assigned a boolean value specifying whether the current cell is ignored while calculating, indicating the cell contains text. TRUE means the cell contains text; FALSE means the cell doesn't contain text.

CellContentsFormula!	110	The variable is assigned a boolean value specifying whether the content of the current cell is a formula. TRUE means the cell contains a formula; FALSE means the cell doesn't contain a formula.
CellLocked!	111	The variable is assigned a boolean value specifying whether the current cell is locked so its contents cannot be modified. TRUE means the cell is locked; FALSE means the cell is not locked.
LeftChar!	112	The variable is assigned a character expression containing the character to the left of the insertion point. If a code, instead of a character, is to the left of the insertion point, the variable is assigned an empty character expression. You can then use this command with the LeftCode! value set member to locate the code to the left of the insertion point.
RightChar!	113	The variable is assigned a character expression containing the character to the right of the insertion point. If a code, instead of a character, is to the right of the insertion point, the variable is assigned an empty character expression. You can then use this command with the RightCode! value set member to locate the code to the right of the insertion point.
MacroPath!	114	This variable contains the path to the WordPerfect macros directory (for example, C:\WPWIN\MACROS\) as specified in Location of Files. (Note the backslash at the end of the path.) This path is assigned to the variable as a character expression.
SharedPath!	115	This variable contains the path to the WordPerfect shared DLL files and program executable files (for example, C:\WPC\). (Note the backslash at the end of the path.) This path is assigned to the variable as a character expression.
BetweenColumnCodes!	116	The variable is assigned a boolean value specifying whether the insertion point is between [Col Def] and [Col Off] codes. TRUE means the insertion point is between column codes; FALSE means the insertion point is not between column codes.

CurrentDocument!	128	The variable is assigned the active document number. The numeric expression has a value from 1 to 9.
AtMainEditScreen!	129	The variable is assigned a boolean value specifying whether an editing window (such as a Text or Figure Editor) other than a document window is active. TRUE means a document window is active; FALSE means an editing window other than a document window is active.
MacroDefActive!	130	The variable is assigned a boolean value specifying whether a macro is currently being recorded. TRUE means a macro is being recorded; FALSE means a macro is not being recorded.
MacroExecuteActive!	131	The variable is assigned a boolean value specifying whether a macro is currently playing, which is always the case. TRUE means a macro is playing; FALSE means a macro is not playing.
MergeActive!	132	The variable is assigned a boolean value specifying whether a merge is currently active. TRUE means a merge is active; FALSE means a merge is not active.
SelectModeActive!	133	The variable is assigned a boolean value specifying whether either Select Mode is active or something is selected. TRUE means Select Mode is active or something is selected; FALSE means Select Mode is not active or nothing is selected.
TypeoverMode!	134	The variable is assigned a boolean value specifying whether Typeover is active. TRUE means Typeover is active; FALSE means Typeover is not active.
RevealCodesActive!	135	The variable is assigned a boolean value specifying whether Reveal Codes is active. TRUE means Reveal Codes is active; FALSE means Reveal Codes is not active.
MajorVersion!	136	The variable is assigned an integer numeric expression specifying the major version number of WordPerfect. For instance, the major version number for WordPerfect 5.1 is 5.
MinorVersion!	137	The variable is assigned an integer numeric expression specifying the minor version number of WordPerfect. For

instance, the minor version number for WordPerfect 5.1 is 1.

InterimRelease!	138	The variable is assigned an integer numeric expression specifying the current interim release number for WordPerfect. For instance, the interim release number for WordPerfect 5.1 is 0. The interim release number for the first interim release of WordPerfect 5.1 is 1, the interim release number for the second interim release is 2, and so forth.
AutoCodePlacement!	139	The variable is assigned a boolean value specifying whether Automatic Code Placement in the Environment Settings dialog box is currently on. TRUE means Automatic Code Placement is on; FALSE means Automatic Code Placement is not on.
ConfirmationOnCodeDelete!	140	The variable is assigned a boolean value specifying whether Confirm on Code Deletion in the Environment Settings dialog box is currently on. TRUE means Confirm on Code Deletion is on; FALSE means Confirm on Code Deletion is not on.
ReadOnlyDoc!	141	The variable is assigned a boolean value specifying whether the active document is read-only (changes to the document cannot be saved under the current document name). TRUE means the document is read-only; FALSE means the document is not read-only.
DocSummaryPromptOnExit!	142	The variable is assigned a boolean value specifying whether the Create Summary on Save/Exit in the Preferences Document Summary dialog box is currently on. TRUE means Create Summary on Save/Exit is on; FALSE means Create Summary on Save/Exit is not on.
SelectedText!	143	The variable is assigned a character expression containing the currently selected text. If no text is selected, the variable is assigned an empty character expression. Only the text in the selection is assigned to the variable. Any codes, including Hard Return and Tab codes, are not included in the character expression.
SelectedText WithHRt!	144	The variable is assigned a character expression containing the currently

selected text. If no text is selected, the variable is assigned an empty character expression. Only the text and Hard Returns in the selection are assigned to the variable. Any codes, except Hard Return codes, are excluded from the character expression.

PrinterType!	145	The variable is assigned a value representing the printer type of the currently selected printer. The values are: 0 - WP printer driver; 1 - Windows printer driver.
PauseKey!	146	The variable is assigned a character expression representing the character that was pressed to continue macro execution after having been paused using the PauseKey macro command.

### WordPerfect Codes

The following are the values and numeric equivalents for WordPerfect codes. Refer to LeftCode! or RightCode! under SystemVariable above for more information.

Value	Code
10	[HRt]
11	[SPg]
12	[HPg]
13	[SRt]
33024	[JustOn]
33280	[Just Off]
34304	[Center Pg]
34560	[ColOn]
35328	[W/O On]
35584	[W/O Off]
35840	[HRt-SPg]
36096	[Note Num]
36352	[Box Num]
36864	[DSRt]
37120	[DSRt] (if on a page break)
37376	[SPg] (not a space or [HRt])
37632	[ISRt]



38656 (R), 38400 (L)	[Select]
39168	[Dorm HRt]
39424	[/] (Cancel Hyphenation)
39680	[End Def]
40448	[Hyph Off]
40704	[Hyph On]
40960	[ ] (Hard Space)
41216	[+]
41472	[t]
41728	[=]
41984	[T]
42240	[*]
42496	[!]
42752	[Math On]
43008	[Math Off]
43264	[-] (Hyphen Character)
44032	- (Soft Hyphen)
44800	[Col Off]
45312	[N]
45568	[Outline Off]
49408	[TAB]
49410	[Tab]
49424	[TAB] (with dot leader)
49426	[Tab] (with dot leader)
49474	[Dec Tab]
49480	[RGT TAB]
49482	[Rgt Tab]
49490	[Dec Tab] (with dot leader)
49496	[RGT TAB] (with dot leader)
49498	[Rgt Tab] (with dot leader)
49504	[Flsh Rgt]
49520	[Flsh Rgt] (with dot leader)
49536	[Mar Rel]
49608	[CNTR TAB]
49610	[Cntr Tab]
49624	[CNTR TAB] (with dot leader)
49626	[Cntr Tab] (with dot leader)

49632	[Center]
49648	[Center] (with dot leader)
49664	[Indent]
49665	[Indent]
49920	[EXT LARGE]
49921	[VRY LARGE]
49922	[LARGE]
49923	[SMALL]
49924	[FINE]
49925	[SUPRSCPT]
49926	[SUBSCPT]
49927	[OUTLN]
49928	[ITALC]
49929	[SHADW]
49930	[REDLN]
49931	[DBL UND]
49932	[BOLD]
49933	[STKOUT]
49934	[UND]
49935	[SM CAP]
50176	[ext large]
50177	[vry large]
50178	[large]
50179	[small]
50180	[fine]
50181	[suprscpt]
50182	[subscpt]
50183	[outln]
50184	[italc]
50185	[shadw]
50186	[redln]
50187	[dbl und]
50188	[bold]
50189	[stkout]
50190	[und]
50191	[sm cap]
50432	[Block Pro:On]

50433	[Block Pro:Off]
53248	[Ln Height]
53249	[L/RMar]
53250	[Ln Spacing]
53251	[HZone]
53252	[Tab Set]
53253	[T/B Mar]
53254	[Just]
53255	[Suppress]
53256	[PgNumbering]
53259	[Paper Sz/Typ]
53504	[Color]
53505	[Font]
53760	[Math Def]
53761	[Col Def]
53762	[Par Num Def]
53763	[Ftn Opt]
53764	[End Opt]
53765	[Fig Opt]
53766	[Tbl Opt]
53767	[Txt Opt]
53768	[Usr Opt]
53769	[Equ Opt]
53771	[Tbl Def]
53773	[Link]
53774	[Link End]
53777	[Brdr Opt]
54016	[Decml/Algn Char]
54017	[Undrln]
54018	[New Ftn Num]
54019	[New End Num]
54020	[Pg Num]
54021	[Ln Num]
54022	[Adv]
54023	[Force]
54024	[BLine]
54026	[Wrd/Ltr Spacing]

54027	[Just Lim]
54028	[New Fig Num]
54029	[New Tbl Num]
54030	[New Txt Num]
54031	[New Usr Num]
54032	[New Equ Num]
54033	[Lang]
54034	[Pg Num Style]
54528	[Header A]
54529	[Header B]
54530	[Footer A]
54531	[Footer B]
54784	[Footnote]
54785	[Endnote]
55040	[Mark]
55041	[End Mark]
55042	[Def Mark]
55043	[Index]
55044	[ToA]
55045	[Endnote Placement]
55047	[Ref]
55048	[Target]
55049	[Subdoc]
55050	[Subdoc Start]
55051	[Subdoc End]
55296	[Date]
55297	[Par Num]
55298	[Ovrstk]
55299	[Insert Pg Num]
55552	[Ptr Cmnd]
55553	[CndI EOP]
55554	[Comment]
55555	[Kern]
55556	[Outline On]
55557	[Leading Adj]
55808	[Fig Box]
55809	[Tbl Box]

55810	[Text Box]
55811	[Usr Box]
55812	[Equ Box]
55813	[HLine]
55814	[VLine]
56064 (R), 56067 (L)	[Style:On] or [Outline Lvl # Style On]
56065 (R), 56067 (L)	[Style:Off] or [Outline Lvl # Style Off]
56066 (R), 56067 (L)	[Open Style] or [Outline Lvl # Open Style]
56320	[Cell]
56321	[Row]
56322	[Tbl Off]
56579	[Hrd Row]
56864	[Mrg:ASSIGN]
56865	[Mrg:BELL]
56866	[Mrg:BREAK]
56867	[Mrg:CALL]
56868	[Mrg:CANCEL OFF]
56869	[Mrg:CANCEL ON]
56870	[Mrg:CASE]
56871	[Mrg:CASE CALL]
56872	[Mrg:CHAIN MACRO]
56873	[Mrg:CHAIN PRIMARY]
56874	[Mrg:CHAIN SECONDARY]
56875	[Mrg:CHAR]
56876	[Mrg:COMMENT]
56877	[Mrg:CTON]
56878	[Mrg:DATE]
56879	[Mrg:DOCUMENT]
56880	[Mrg:ELSE]
56881	[Mrg:END FIELD]
56882	[Mrg:END FOR]
56883	[Mrg:END IF]
56884	[Mrg:END RECORD]
56885	[Mrg:END WHILE]
56886	[Mrg:FIELD]
56887	[Mrg:FOR]
56889	[Mrg:GO]

56890	[Mrg:IF]
56891	[Mrg:IF BLANK]
56892	[Mrg:IF EXISTS]
56893	[Mrg:IF NOT BLANK]
56894	[Mrg:KEYBOARD]
56895	[Mrg:LABEL]
56896	[Mrg:LOCAL]
56897	[Mrg:LOOK]
56898	[Mrg:MID]
56899	[Mrg:MRG CMND]
56900	[Mrg:NEST MACRO]
56901	[Mrg:NEST PRIMARY]
56902	[Mrg:NEST SECONDARY]
56903	[Mrg:NEXT]
56904	[Mrg:NEXT RECORD]
56905	[Mrg:NTOC]
56906	[Mrg:PROCESS]
56907	[Mrg:ON CANCEL]
56908	[Mrg:ON ERROR]
56909	[Mrg:PAGE OFF]
56910	[Mrg:PAGE ON]
56911	[Mrg:PRINT]
56912	[Mrg:PROMPT]
56913	[Mrg:QUIT]
56914	[Mrg:RETURN]
56915	[Mrg:RETURN CANCEL]
56916	[Mrg:RETURN ERROR]
56917	[Mrg:REWRITE]
56918	[Mrg:STEP OFF]
56919	[Mrg:STEP ON]
56920	[Mrg:SUBST PRIMARY]
56921	[Mrg:SUBST SECONDARY]
56922	[Mrg:SYSTEM]
56923	[Mrg:TEXT]
56924	[Mrg:VARIABLE]
56925	[Mrg:WAIT]
56926	[Mrg:WHILE]

56927	[Mrg:STATUS PROMPT]
56928	[Mrg:INPUT]
56929	[Mrg:LEN]
56930	[Mrg:FIELD NAME]
56931	[Mrg:STOP]
56932	[Mrg:OTHERWISE]

Show Me!

---

## Grammatik

The Grammatik command invokes the Grammatik style and grammar checker for the highlighted text or for the entire document.

### **syntax**

Grammatik (FileName: *Character Expression*)

### **parameters**

This command has one parameter.

#### **FileName**

This parameter is a character expression. The FileName specifies a file in which to execute a grammar check. This parameter is optional.

Speller

Thesaurus



---

## GraphicLine

Use the GraphicLine command to create horizontal or vertical lines, or to edit a selected horizontal or vertical line. This command is the equivalent of setting options in the Create Horizontal Line, Create Vertical Line, Edit Horizontal Line, or Edit Vertical Line [dialog box](#).

### syntax

GraphicLine (Operation:*Value Set Member*; LineLength:*Measurement Expression*; Thickness:*Measurement Expression*; Shading:*Numeric Expression*; VerticalType:*Value Set Member*; VerticalPosition:*Measurement Expression*; HorizontalType:*Value Set Member*; HorizontalPosition:*Measurement Expression*; Column:*Numeric Expression*)

### parameters

This command has nine parameters:

#### Operation:

This parameter specifies whether a horizontal or vertical line is being created, or whether a selected graphic line is being modified. The members of this value set are:

Member	Numeric Equivalent	Definition
HCreate!	0	A horizontal line is being created.
VCreate!	1	A vertical line is being created.
Edit!	2	A selected line is to be modified.

#### LineLength:

This measurement expression specifies the length of the line being created or edited. This parameter is ignored and can be omitted if the macro is creating or editing a vertical line and the VerticalType parameter is set to FullPage!. It can also be omitted if the macro is creating or editing a horizontal line and the HorizontalType parameter is set to Full!.

#### Thickness:

This measurement expression specifies the thickness of the line being created or edited.

#### Shading:

This numeric expression specifies the gray shading percentage for the line being created. The higher the number, the darker the line: at 100, the line is black. This numeric expression should have a value from 0 to 100.

#### VerticalType:

This parameter specifies the vertical location of the line being created or edited. The members of this value set are:

Member	Numeric Equivalent	Description
Baseline!	0	The line appears on the baseline of the current line. This member is only valid if the line being created or edited is horizontal.
FullPage!	1	The line runs from the top margin to the bottom margin. This member is only valid if the line being created or edited is vertical. If the VerticalType parameter is set to this value, the LineLength parameter is ignored and can be omitted.

Top!	2	The line begins at the top margin and extends down. This member is only valid if the line being created or edited is vertical.
Center!	3	The line is centered between the top and bottom margins. This member is only valid if the line being created or edited is vertical.
Bottom!	4	The line begins at the bottom margin and extends up. This member is only valid if the line being created or edited is vertical.
Specify!	5	The line begins at the vertical point specified in the VerticalPosition parameter.

**VerticalPosition:**

This measurement expression specifies the vertical beginning point for the line being created or edited. If the VerticalType parameter is not set to Specify!, this parameter is ignored and can be omitted.

**HorizontalType:**

This parameter specifies the horizontal location of the line being created or edited. The members of this value set are:

Member	Numeric Equivalent	Description
Left!	0	The line goes down the left margin if the line being created or edited is vertical. If the line is horizontal, the line begins at the left margin and extends to the right.
Right!	1	The line goes down the right margin if the line being created or edited is vertical. If the line is horizontal, the line begins at the right margin and extends to the left.
Center!	2	The line is centered between the left and right margins. This member is only valid if the line being created or edited is horizontal.
Full!	3	The line runs from the left margin to the right margin. This member is only valid if the line being created or edited is horizontal. If the HorizontalType parameter is set to this value, the LineLength parameter is ignored and can be omitted.
Columns!	4	The line is placed to the right of the column specified in the Column parameter. If the HorizontalType parameter is not set to Columns!, the Column parameter is ignored and can be omitted.
Specify!	5	The line begins at the horizontal point specified in the HorizontalPosition parameter.

**HorizontalPosition:**

This measurement expression specifies the horizontal beginning point for the line being created or edited. If the HorizontalType parameter is not set to Specify!, this parameter is

ignored and can be omitted.

**Column:**

This parameter specifies a column number. The vertical line being created or edited is placed to the right of that column. When a vertical line is being created, the first column is counted "0", not "1".

This parameter is not valid if a horizontal line is being created or edited, or if the HorizontalType parameter is not set to Columns!.

GraphicLineSelect

HorizLineEditDlg

VertLineEditDlg

---

## GraphicLineSelect

The GraphicLineSelect command selects the first line (of the type specified in the Type parameter) it encounters. WordPerfect uses a certain method to decide which line to select. First it searches from the insertion point backwards until it finds the specified type of graphic line code. If it does not find one, it searches forward to the first graphic line code of the specified type in the document.

### **syntax**

GraphicLineSelect (Type:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **Type:**

This parameter specifies whether WordPerfect should select a vertical or horizontal line. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Horizontal!	0
-------------	---

Vertical!	1
-----------	---

GraphicLine

---

## GraphicsShow

Use this command to specify whether the contents of graphics boxes will display in WordPerfect. This command is the equivalent of checking or unchecking Graphics from the View menu.

### **syntax**

GraphicsShow (State:*Value Set Member*)

### **parameters**

This command has one parameter. You may omit this parameter if you want the GraphicsShow command to switch to the opposite of the current graphic display state.

### **State**

This parameter specifies whether graphics will be displayed. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

---

## HardHyphen

The HardHyphen command inserts a hyphen into the active document at the insertion point. This command is the equivalent of either choosing hyphen in the Insert Special Codes dialog box, then choosing Insert, or simply pressing the hyphen key (-).

### **syntax**

HardHyphen ( )

InsertSpecialCodesDlg

---

## HardPageBreak

The HardPageBreak command inserts a new page (a Hard Page code ([HPg])) at the insertion point in the active document. This command is the equivalent of pressing Ctrl+Enter.

### **syntax**

HardPageBreak ( )

---

## HardPageBreakInsert

Use the HardPageBreakInsert command with the Outline feature to end the current page without inserting the paragraph number.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

HardPageBreakInsert ( )

[HardPageBreak](#)



---

## HardReturn

The HardReturn command ends the current line at the insertion point. The HardReturn command is the equivalent of pressing Enter or Return in a text editing window.

### **syntax**

HardReturn ( )

HardReturnInsert

---

## HardReturnInsert

Use the HardReturnInsert command with the Outline feature to end the current line without inserting a paragraph number. This command is the equivalent of pressing Shift+Enter.

### **syntax**

HardReturnInsert ( )

HardReturn

---

## HardSpace

The HardSpace command inserts a character that looks like a space into the active document. However, this character, unlike a normal space, is not treated as a word break and cannot be used as a place to wrap text to the next line. Essentially, two words connected by a hard space are treated as a single word by WordPerfect. This is useful if you always want two words, like a name or date, to be kept together on the same line. The HardSpace command is the equivalent of pressing Ctrl+Space.

### **syntax**

HardSpace ( )

---

## HardTabCenter

The HardTabCenter command inserts a Center Tab code ([HdCntrTab]) into the active document at the insertion point. This code causes text typed at the next tab stop to be centered, regardless of the original tab setting type. This command is the equivalent of choosing the Center radio button in the Special Codes dialog box.

### **syntax**

HardTabCenter ( )

---

## HardTabCenterDot

The HardTabCenterDot command inserts a Center Dot Leader Tab code ([HdCntrTab]) into the active document at the insertion point. This code causes text typed at the next tab stop to be centered with a dot leader, regardless of the original tab setting type. This command is the equivalent of choosing the Center with Dot Leader radio button in the Special Codes dialog box.

### **syntax**

HardTabCenterDot ( )

InsertSpecialCodesDlg

---

## HardTabDecimal

The HardTabDecimal command inserts a Decimal Tab code ([HdDecTab]) into the active document at the insertion point, unless there are no more tab stops available on the current line. This code causes text typed at the next tab stop to be centered on the decimal, regardless of the original tab setting type.

### **syntax**

HardTabDecimal ( )

[HardTabDecimalInsert](#)

[HardTabDecimalDot](#)

[InsertSpecialCodesDlg](#)

---

## HardTabDecimalDot

The HardTabDecimalDot command inserts a Decimal Dot Tab code ([HdDecTab]) into the active document at the insertion point. This code causes text typed at the next tab stop to have a dot leader and to be centered on the decimal, regardless of the original tab setting type. This command is the equivalent of choosing the Decimal [HdDecTab] radio button in the Special Codes dialog box.

### **syntax**

HardTabDecimalDot ( )

InsertSpecialCodesDlg

HardTabDecimal

---

## HardTabDecimalInsert

The HardTabDecimalInsert command inserts a Decimal Tab code ([HdDecTab]) into the active document at the insertion point, even if there are no more tab stops available on the current line. This code causes text typed at the next tab stop to be centered on the decimal, regardless of the original tab setting type. If no decimal is entered, all text typed at this tab stop is aligned to the left of the tab stop. This command is the equivalent of choosing the Decimal Align radio button in the Special Codes dialog box.

### **syntax**

HardTabDecimalInsert ( )

InsertSpecialCodesDlg

HardTabDecimal



---

## HardTabLeft

The HardTabLeft command inserts a Left Tab code ([HdTab]) into the active document at the insertion point. This code causes text typed at the next tab stop to be left aligned, regardless of the original tab setting. This command is the equivalent of choosing the Left Align radio button in the Special Codes dialog box.

### **syntax**

HardTabLeft ( )

InsertSpecialCodesDlg

HardTabLeftDot

---

## HardTabLeftDot

The HardTabLeftDot command inserts a Tab Left Dot code ([HdTab]) into the active document at the insertion point. This code causes text typed at the next tab stop to have a dot leader and to be left-aligned, regardless of the original tab setting type. This command is the equivalent of choosing the Left Align with Dot Leader radio button in the Special Codes dialog box.

### **syntax**

HardTabLeftDot ( )

InsertSpecialCodesDlg

HardTabLeft

---

## HardTabRight

The HardTabRight command inserts a Right Tab code ([HdRgtTab]) into the active document at the insertion point. This code causes text typed at the next tab stop to be right aligned, regardless of the original tab setting type. This command is the equivalent of choosing the Right Align radio button in the Special Codes dialog box.

### **syntax**

HardTabRight ( )

InsertSpecialCodesDlg

HardTabRightDot

---

## HardTabRightDot

The HardTabRightDot command inserts a Right Dot Leader Tab code ([HdRgtTab]) into the active document at the insertion point. This code causes text typed at the next tab stop to have a dot leader and to be right-aligned, regardless of the original tab setting type. This command is the equivalent of choosing the Right Align with Dot Leader radio button in the Special Codes dialog box.

### **syntax**

HardTabRightDot ( )

InsertSpecialCodesDlg

HardTabRight

---

## HeaderDlg

The HeaderDlg command displays the Headers dialog box, which lets the user specify whether to create, edit, or discontinue a specified header. This command is the equivalent of choosing Page from the Layout menu, then choosing Headers.

For information on using dialog boxes with macros, see Dialog Boxes and Macros under Using Product Commands in Using Macros.

### **syntax**

HeaderDlg ( )

HeaderFooter

---

## HeaderFooter

Use the HeaderFooter command to create, edit, or discontinue headers and footers. This command is the equivalent of setting options in the Headers and Footers dialog boxes.

### **syntax**

HeaderFooter (Operation:*Value Set Member*; Item:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **Operation:**

This parameter specifies whether the macro should create, edit, or discontinue the header or footer specified in the Item parameter of this command. The values of this parameter set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Create!	0
Discontinue!	1
Edit!	2

Both the Create and Edit parameter members open a header or footer editing window. Once the editing window is open, subsequent commands can insert text and [codes](#) into the header or footer. A Close command will close this editing window and, if Auto Code Placement is on, a Header or Footer code will be placed at the beginning of the current page of the active document. If Auto Code Placement is off, a Header or Footer code will be placed at the insertion point of the active document.

#### **Item:**

This parameter specifies which header or footer will be created, edited, or discontinued. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

HeaderA!	0
HeaderB!	1
FooterA!	2
FooterB!	3

[Close](#)

[FooterDlg](#)

[HeaderDlg](#)

---

## HeaderFooterPlacement

The HeaderFooterPlacement command specifies whether the header or footer currently being edited will be placed on odd pages, even pages, or all pages. This command is the equivalent of choosing Placement in a header or footer editor, then choosing Every Page, Odd Pages, or Even Pages. This command should only be encountered when a header or footer editor is active.

### **syntax**

HeaderFooterPlacement (State: *Value Set Member*)

### **parameters**

This command has one parameter:

#### **State:**

This parameter specifies the pages on which the header or footer being edited will be printed. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

EveryPage!	0
------------	---

EvenPages!	1
------------	---

OddPages!	2
-----------	---

[HeaderFooter](#)

[HeaderFooterPlacementDlg](#)

---

## HeaderFooterPlacementDlg

The HeaderFooterPlacementDlg command displays the Placement dialog box, which lets the user specify whether the header or footer currently being edited will be placed on odd pages, even pages, or all pages. This command is the equivalent of choosing Placement in a header or footer editor. This command should only be encountered when a header or footer editor is active.

### **syntax**

HeaderFooterPlacementDlg ( )

HeaderFooter

HeaderFooterPlacement



---

## HelpContextSensitive

The HelpContextSensitive command opens the Help system and displays context-sensitive help on the currently active item. For instance, if Print Preview is active when the HelpContextSensitive command is encountered, the Help system displays help on Print Preview. The HelpContextSensitive command is the equivalent of pressing F1.

The macro does not pause when it encounters this command. Instead, the macro continues playing. If you want the macro to pause, use the PAUSE programming command.

### **syntax**

HelpContextSensitive ( )

[HelpGlossary](#)

[HelpHowDoI](#)

[HelpWhatIs](#)

---

## HelpGlossary

The HelpGlossary command opens the Help system and displays a list of WordPerfect and Windows terms. This command is the equivalent of choosing Glossary from the Help menu.

The macro does not pause when it encounters this command. Instead, the macro continues playing. If you want the macro to pause, use the PAUSE programming command.

### **syntax**

HelpGlossary ( )

[HelpContextSensitive](#)

[HelpKeyboard](#)

[HelpUsingWPHelp](#)

---

## HelpHowDol

The HelpHowDol command opens the Help system and explains how to perform common tasks. This command is the equivalent of choosing How Do I from the Help [menu](#).

The macro does not pause when it encounters this command. Instead, the macro continues playing. If you want the macro to pause, use the PAUSE programming command.

### **syntax**

HelpHowDol ( )

[HelpContextSensitive](#)

[HelpIndex](#)

[HelpUsingWPHelp](#)

[HelpWhatIs](#)

---

## HelpIndex

The HelpIndex command brings up the table of contents of the WordPerfect help file. This command is the equivalent of choosing Contents from the Help menu.

The macro does not pause when it encounters this command. Instead, the macro continues playing. If you want the macro to pause, use the PAUSE programming command.

### **syntax**

HelpIndex ( )

[HelpGlossary](#)

[HelpUsingWPHelp](#)

[HelpWhatIs](#)

[HelpUsingMacros](#)

---

## HelpKeyboard

The HelpKeyboard command opens the Help system, which provides information about keyboard remapping and displays templates for the WordPerfect CUA (Common User Access) keyboard and the WordPerfect for DOS compatible keyboard. This command is the equivalent of choosing Keyboard from the Help menu.

The macro does not pause when it encounters this command. Instead, the macro continues playing. If you want the macro to pause, use the PAUSE programming command.

### **syntax**

HelpKeyboard ( )

[HelpContextSensitive](#)

[HelpGlossary](#)

[HelpWhatIs](#)

---

## HelpUsingMacros

This command brings up the Macro Help File (what you are searching through now) and is the equivalent of choosing Using Macros from Help on the main menu.

### **syntax**

UsingMacroHelp ( )

[HelpIndex](#)

[HelpUsingWPHelp](#)

---

## HelpUsingWPHelp

The HelpUsingWPHelp command opens the Help system, then describes how to use the Help system. This command is the equivalent of choosing Using Help from the Help menu.

The macro does not pause when it encounters this command. Instead, the macro continues playing. If you want the macro to pause, use the PAUSE programming command.

### **syntax**

HelpUsingWPHelp ( )

[HelpContextSensitive](#)

[HelpGlossary](#)

[HelpHowDoI](#)

[HelpIndex](#)

[HelpKeyboard](#)

[HelpWhatIs](#)

[HelpUsingMacros](#)

---

## HelpWhatIs

The HelpWhatIs command opens the Help system so the user can get Help by pointing with the mouse or pressing keystroke combinations. This command is the equivalent of choosing What Is from the Help menu.

The macro does not pause when it encounters this command. Instead, the macro continues playing. If you want the macro to pause, use the PAUSE programming command.

### **syntax**

HelpWhatIs ( )

[HelpContextSensitive](#)

[HelpHowDoI](#)

[HelpIndex](#)



---

## HorizLineCreateDlg

The HorizLineCreateDlg command displays the Create Horizontal Line dialog box, which lets the user create a horizontal line in the active document and specify the line size, position, and shading. The HorizLineCreateDlg command is the equivalent of choosing Line from the Graphics menu, then choosing Horizontal.

### **syntax**

HorizLineCreateDlg ( )

GraphicLine

GraphicLineSelect

HorizLineEditDlg

VertLineCreateDlg

---

## HorizLineEditDlg

The HorizLineEditDlg command displays the Edit Horizontal Line dialog box, which lets the user edit a selected horizontal line. If a horizontal line is already selected when the HorizLineEditDlg command is encountered, the dialog box lets the user edit that line. If a horizontal line is not selected when this command is encountered, WordPerfect searches for the horizontal line to be edited. From the insertion point, WordPerfect first searches backwards. If it finds a Horizontal Line code, that line is edited in the Edit Horizontal Line dialog box. If no horizontal line is found when searching backward, WordPerfect edits the first horizontal line in the document. If no horizontal line is found in any direction, WordPerfect ignores the HorizLineEditDlg command.

The HorizLineEditDlg command is the equivalent of choosing Line from the Graphics menu, then choosing Edit Horizontal.

### **syntax**

HorizLineEditDlg ( )

GraphicLine

GraphicLineSelect

HorizLineCreateDlg

VertLineCreateDlg

---

## HorizScrollBarShow

The HorizScrollBarShow command allows you to determine whether or not the horizontal scroll bar will be displayed. It is the equivalent of opening the Preferences Display dialog box and clicking in the HorizScrollBar checkbox or choosing Horizontal Scroll from the View menu.

### **syntax**

HorizScrollBarShow (State: *Value Set Member*)

### **parameter:**

This command has one parameter.

### **State:**

This parameter specifies whether or not the horizontal scroll bar will be shown. The members and corresponding numeric equivalents of the State value are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

---

## Hyphen

The Hyphen command inserts a dash character into the active document at the insertion point. This command is the equivalent of choosing Dash Character in the Insert Special Codes dialog box, then choosing Insert.

### **syntax**

Hyphen ( )

InsertSpecialCodesDlg

---

## HyphenIgnoreWord

The HyphenIgnoreWord command inserts a Hyphen Ignore Word code ([Hyph Ign Wrd]) at the insertion point so hyphenation is canceled for the current word. This command is the equivalent of choosing Hyphenation Ignore Word, then choosing Insert from the Insert Special Codes dialog box.

### **syntax**

HyphenIgnoreWord ( )

InsertSpecialCodesDlg

---

## HyphenSoft

The HyphenSoft command inserts a Soft Hyphen code into the active document at the insertion point. The soft hyphen divides a word and displays a hyphen when the word falls at the end of a line. If the word does not need to be hyphenated, the Soft Hyphen code remains in the document codes but has no effect. This command is the equivalent of choosing Soft Hyphen, then choosing Insert in the Insert Special Codes dialog box.

### **syntax**

HyphenSoft ( )

InsertSpecialCodesDlg

---

## HyphenSoftReturn

The HyphenSoftReturn command inserts a Hyphen Soft Return code ([HyphSRt]) that tells WordPerfect where a word should be broken if the word needs to be hyphenated. In other words, this code is like a Soft Hyphen code, except it doesn't produce a hyphen--it simply breaks the word if that word falls near the end of a line. This command is the equivalent of choosing Hyphenation Soft Return, then Insert from the Insert Special Codes dialog box.

### **syntax**

HyphenSoftReturn ( )

HyphenSoft

InsertSpecialCodesDlg

---

## IF

The IF command is the opening command for the IF-ENDIF conditional statement. WordPerfect runs all commands between the IF and ENDIF commands when the Test parameter is true. If the Test parameter is false, the macro continues from the statement following the ENDIF statement.

If you want certain commands to be used only when the expression is not true, use the ELSE command. (See ELSE.)

### **syntax**

IF (*Test*)

### **parameters**

This command has one parameter:

#### **Test**

This parameter is a relational expression--an expression that can be evaluated as true or false. For instance, NUMBER>34 could be either true or false, depending on the value of variable NUMBER, so it is a valid relational expression.

ENDIF

ELSE

Show Me!



---

## **INTEGER**

The INTEGER command takes the integer portion of the Numeric Expression parameter (usually a mathematical equation) and assigns it to the variable. For instance, if a macro contains INTEGER (VAR01; 43/2), the value 21 will be assigned to VAR01. The fractional portion (.5) is discarded. If the numeric expression does not contain an integer, the variable is set to 0. If the numeric expression is a negative number, this command rounds the number (for example, -123.45 becomes -124).

### **syntax**

INTEGER (*Variable; Numeric Expression*)

### **parameters**

This command has two parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Numeric Expression**

Any numeric expression is a valid value for this parameter. Usually, this numeric expression contains some sort of mathematical operation.

## FRACTION

Show Me!

---

## IndexDefine

The IndexDefine command inserts an Index Definition code ([Def Mark:Index]) into the active document at the insertion point. This code indicates where the index will be located when the document is generated. The IndexDefine command sets the index numbering style and the name of the concordance file. This command is the equivalent of setting options in the Define Index dialog box.

### **syntax**

IndexDefine (NumberingStyle:*Value Set Member*; ConcordanceFile:*Character Expression*)

### **parameters**

This command has two parameters:

#### **NumberingStyle:**

This parameter specifies how the text and page numbers in the index will look. The members and corresponding numeric equivalents of this value set correspond to the Numbering Format pop-up list in the Define Index dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

NoNumbering!	0
--------------	---

NumFollowsEntry!	1
------------------	---

FollowsInParens!	2
------------------	---

FlushRight!	3
-------------	---

FlushLeaders!	4
---------------	---

#### **ConcordanceFile:**

This character expression specifies the name or full path of the file containing the index concordance. You do not need to include this parameter unless a concordance will be used with the index.

IndexDefineDlg

---

## **IndexDefineDlg**

The IndexDefineDlg command displays the Define Index dialog box, which lets the user specify a concordance file and the numbering format for the index. When the user chooses OK, a Define Index code is placed at the insertion point in the active document. The IndexDefineDlg command is the equivalent of choosing Define from the Tools menu, then choosing Index.

### **syntax**

IndexDefineDlg ( )

IndexDefine

---

## IndexMark

The IndexMark command marks a point in the active document to be referenced in the index and specifies the heading and (optionally) the subheading of the index reference. When you generate the index, an entry with the specified heading and subheading will be included in the index. This command is the equivalent of filling out the Mark Index dialog box.

### **syntax**

IndexMark (HeadingText:*Character Expression*; SubheadingText:*Character Expression*)

### **parameters**

This command has two parameters:

#### **HeadingText:**

This character expression specifies the main heading for the index entry being created.

#### **SubheadingText:**

This character expression specifies the subheading for the index entry being created. You can omit this parameter if you don't want a subheading.

IndexDefine

IndexMarkDlg

---

## **IndexMarkDlg**

The IndexMarkDlg command displays the Mark Index dialog box, which lets the user set an index heading and subheading mark for the word at the insertion point. This command is the equivalent of choosing Mark Text from the Tools menu, then choosing Index.

### **syntax**

IndexMarkDlg ( )

IndexMark

---

## InhibitInput

The InhibitInput command allows you to prevent or allow keyboard and menu input during the execution of a macro. The default setting is Off, which allows input.

### **syntax**

InhibitInput (State: Value Set Member)

### **parameters**

This command has one parameter.

### **State**

This parameter is a value set member. The State parameter acts as a "toggle" to allow or prevent keyboard and menu input during the execution of a macro. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Description</b>
Off!	0	Allows keyboard and menu input.
On!	1	Prevents keyboard and menu input.

Display

---

## InsertSpecialCodesDlg

The InsertSpecialCodesDlg command displays the Insert Special Codes dialog box, which lets the user insert the different types of Hard Tab, Hyphenation, and other codes into the active document at the insertion point. This command is the equivalent of choosing Line from the Layout menu, then choosing Special Codes.

### **syntax**

InsertSpecialCodesDlg ( )

---

## InsertTypeover

Use the InsertTypeover command to turn WordPerfect Typeover mode on or off. "Typeover mode" means that new text will overwrite old text as you type. This command is the equivalent of pressing Insert.

### **syntax**

InsertTypeover (State:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **State:**

This parameter specifies whether Typeover is on or off. This parameter must be specified or an error will result. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---



---

## JustifyCenter

The JustifyCenter command causes all text from the current paragraph forward to be centered horizontally between the margins. All text in the document will be centered until a different justification code (such as Left, Right, or Full) is encountered. This command is the equivalent of choosing Justification from the Layout menu, then choosing Center.

### **syntax**

JustifyCenter ( )

JustifyFull

JustifyLeft

JustifyRight

---

## JustifyFull

The JustifyFull command causes all text from the current paragraph forward to be aligned along the left and right margins. All text will be both left- and right-justified until a different justification code (such as Left, Right, or Center) is encountered. This command is the equivalent of choosing Justification from the Layout menu, then choosing Full.

### **syntax**

JustifyFull ( )

JustifyCenter

JustifyLeft

JustifyRight

---

## JustifyLeft

The JustifyLeft command causes all text from the current paragraph forward to be aligned along the left margin. All text will be left-justified until a different justification code (such as Center, Full, or Right) is encountered. This command is the equivalent of choosing Justification from the Layout menu, then choosing Left.

### **syntax**

JustifyLeft ( )

JustifyCenter

JustifyFull

JustifyRight

---

## **JustifyRight**

The JustifyRight command causes all text from the current paragraph forward to be aligned along the right margin. All text will be right-justified until a different justification code (such as Center, Left, or Full) is encountered. This command is the equivalent of choosing Justification from the Layout menu, then choosing Right.

### **syntax**

JustifyRight ( )

JustifyCenter

JustifyFull

JustifyLeft

---

## **KerningSpacing**

The KerningSpacing command moves the character on the right of the insertion point a specified amount of space to the left or right. This command is the equivalent of setting an amount in the Amount text box of the Manual Kerning dialog box.

### **syntax**

KerningSpacing (KerningSpacingValue:*Measurement Expression*)

### **parameters**

This command has one parameter:

#### **KerningSpacingValue**

This measurement expression specifies how much to move the character. If this parameter has a negative value, the character will be moved that much to the left. If the parameter has a positive value, the character will be moved that much to the right.

Advance

---

## KeyboardSelect

The KeyboardSelect command specifies the filename of a "soft keyboard" mapping for WordPerfect to use. The filename can include a path. This command is the equivalent of selecting a keyboard in the Select Keyboard File dialog box.

### **syntax**

KeyboardSelect (KeyboardFile:*Character Expression*)

### **parameters**

This command has one parameter:

#### **KeyboardFile:**

This character expression specifies the filename of the keyboard mapping you want to use with WordPerfect. If you want to select CUA keyboard, use "\*" for the character expression in the KeyboardFile parameter.

KeyboardSelectDlg

---

## KeyboardSelectDlg

The KeyboardSelectDlg command displays the Select Keyboard File dialog box, which lets the user specify a keyboard file to use with WordPerfect. This command is the equivalent of choosing Preferences from the File menu and choosing Keyboard.

This command is not recordable. To use this command you must add it by editing the macro.

### **syntax**

KeyboardSelectDlg ( )

KeyboardSelect

---

## **LABEL**

The LABEL command marks a place in the macro. You can use another command (for instance, CALL, CASE, CASE CALL, GO, ONCANCEL, ONERROR, or ONNOTFOUND) to send macro execution to a LABEL command.

You can have as many LABEL commands in a macro as you like. Be certain, however, to have a unique Name parameter for each LABEL command.

The "at" symbol (@) follows the label name. This symbol delimits the label name and lets the macro know that the text should be treated as a label, not as a character string or variable. The label name should not exceed 15 bytes. (Each standard character is one byte in length; extended characters are often longer than one byte.)

### **syntax**

LABEL (*Name*@)

### **parameters**

This command has one parameter:

#### **Name**

Any text beginning with a letter followed by any number of alphanumeric characters and ending with the "at" symbol (@) is a valid value for this parameter. The label should not be enclosed in quotes since it is not treated as a character string. Be certain to place the at symbol (@) after the label name.

[Show Me!](#)



---

## Language

The Language command inserts a specified language code into the active document at the insertion point, so that the selected language and other default settings associated with that language will be active from that point forward. This command is the equivalent of choosing a language from the Language dialog box.

### **syntax**

Language (Language:*Character Expression*)

### **parameters**

This command has one parameter:

#### **Language:**

This character expression specifies which language should be used for inserting dates and number separators (such as the comma used to separate thousands). This parameter should be two characters long, and should correspond to one of the WordPerfect language abbreviations listed in the Language dialog box.

LanguageDlg

---

## LanguageDlg

The LanguageDlg command displays the Language dialog box, which lets the user specify which language should be used for inserting dates and number separators into WordPerfect documents. This command is the equivalent of choosing Language from the Tools menu.

### **syntax**

LanguageDlg ( )

Language



---

## LastOpenedAdd

The LastOpenedAdd command allows the user to specify a file to be added to the last four opened documents list. The last four opened documents list appears at the bottom of the File menu.

### **syntax**

LastOpenedAdd (FileName: *Character Expression*; MenuText: *Character Expression*)

### **parameters**

This command has two parameters.

#### **FileName**

This parameter is a character expression. The FileName is the name of the file to be added as the last (most recent) file opened.

#### **MenuText**

This parameter is a character expression. The MenuText allows you to specify what text will appear on the File menu with the other most recent files opened.

[FileOpen](#)

---

## LineCenter

The LineCenter command centers all text from the insertion point forward on the current line. If there is no text following the insertion point on the current line, the insertion point moves to the center of the line, and any text then typed on that line is centered. This command applies only to the line containing the insertion point. The LineCenter command is the equivalent of choosing Line from the Layout menu, then choosing Center.

### **syntax**

LineCenter ( )

JustifyCenter

---

## LineCenterEnd

The LineCenterEnd command turns off center justification in the current line without inserting a Hard Return code ([HRT]). Text is centered between the Center Justification code ([Just:Center]) and the End of Centering Alignment code ([End C/A]). This command discontinues decimal alignment and overrides a regular [Center] code. Text after the End of Centering Alignment code is not centered. This command is the equivalent of choosing End Centering/Alignment [End C/A], then choosing Insert from the Insert Special Codes dialog box.

### **syntax**

LineCenterEnd( )

InsertSpecialCodesDlg

---

## LineDraw

The LineDraw command draws a line of a specified length in a specified direction, using a specified line type or character. This command is the equivalent of choosing Line Draw from the Tools menu.

### **syntax**

LineDraw (Item:*Value Set Member*; Direction:*Value Set Member*; Character:*Character Expression*; RepeatCount:*Numeric Expression*)

### **parameters**

This command has four parameters:

#### **Item:**

This parameter specifies the thickness and pattern of the line. It can also specify the Line Draw mode (whether a line will be drawn, the cursor will be moved, or a line will be erased). The following members of this value set correspond with the available line types in the Line Draw dialog box:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

ItemLineSingle!	0
ItemLineDouble!	1
ItemBlockHatchedLeft!	2
ItemBlockHatched!	3
ItemBlockHatchedRight!	4
ItemBlockSolid!	5
ItemBlockSolidBottom!	6.
ItemBlockSolidLeft!	7
ItemBlockSolidRight!	8
ItemBlockSolidTop!	9
ItemCharacter!	10
ItemMove!	11
ItemErase!	12
ItemEnd!	13

#### **Direction:**

This parameter specifies the direction in which the line will be drawn. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

PosCharPrev!	0
PosCharNext!	1
PosLineUp!	2
PosLineDown!	3
PosWordPrev!	4

PosWordNext!	5
PosParagraphNext!	6
PosParagraphPrev!	7
PosLineBegin!	8
PosLineEnd!	9

**Character:**

This character expression specifies a certain character to be used in drawing the line. The character expression used in this parameter should be one character in length. This parameter is ignored unless ItemCharacter is used in the Item parameter.

**RepeatCount:**

This numeric expression determines how many characters long to make the line. This parameter may be omitted if the Direction parameter is set to any of the members or numeric equivalents listed above. If this parameter is missing, the default is 1.

LineDrawDlg



---

## LineDrawDlg

The LineDrawDlg command displays the Line Draw dialog box, which lets the user draw lines with characters by specifying the draw mode and type of character with which to draw. This command is the equivalent of choosing Line Draw from the Tools menu.

### **syntax**

LineDrawDlg ( )

LineDraw

---

## **LineFlushRight**

The LineFlushRight command right-aligns all text that is located on the right side of the insertion point on the current line. If there is no text following the insertion point on the current line, the insertion point moves to the right margin, and any text then typed on that line is aligned on the right margin. This command applies only to the line containing the insertion point. The LineFlushRight command is the equivalent of choosing Line from the Layout menu, then choosing Flush Right.

### **syntax**

LineFlushRight ( )

JustifyRight

---

## LineHeight

The LineHeight command specifies whether the height of a line should be automatically calculated by WordPerfect or set to a fixed height, regardless of the font size. If you want to set a fixed line height, you can also use this command to specify the height. The LineHeight command is the equivalent of setting options in the LineHeight [dialog box](#).

### **syntax**

LineHeight (Type:*Value Set Member* ; Height:*Measurement Expression*)

### **parameters**

This command has two parameters:

#### **Type:**

This parameter specifies whether line height will be set automatically or set to a fixed height. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Auto!	0
-------	---

Fixed!	1
--------	---

#### **Height:**

This measurement expression specifies line height for fixed-height lines. The Height parameter may be omitted if the Type parameter is set to Auto!.

[LineHeightDlg](#)

---

## LineHeightDlg

The LineHeightDlg command displays the Line Height dialog box, which lets the user specify whether line height should be automatically set by WordPerfect or set to a specific height, regardless of the font size. This command is the equivalent of choosing Line from the Layout menu, then choosing Height.

### **syntax**

LineHeightDlg ( )

LineHeight

---

## LineHyphenation

The LineHyphenation command specifies the left and right percentages for the hyphenation zone and whether hyphenation should be turned on. This command is the equivalent of setting options in the Line Hyphenation [dialog box](#).

### **syntax**

LineHyphenation (State:*Value Set Member*; PercentLeft:*Numeric Expression*;  
PercentRight:*Numeric Expression*)

### **parameters**

This command has three parameters:

#### **State:**

This parameter specifies whether hyphenation is turned on or off. The valid members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

#### **PercentLeft:**

This numeric expression specifies the percentage for the left side of the hyphenation zone. This parameter should have a value from 0 to 100.

#### **PercentRight:**

This numeric expression specifies the percentage for the right side of the hyphenation zone. This parameter should have a value from 0 to 100.

[LineHyphenationDlg](#)

---

## LineHyphenationDlg

The LineHyphenationDlg command displays the Line Hyphenation dialog box, which lets the user turn automatic hyphenation on or off and specify the left and right percentages of the hyphenation zone. The LineHyphenationDlg command is the equivalent of choosing Line from the Layout menu, then choosing Hyphenation.

### **syntax**

LineHyphenationDlg ( )

LineHyphenation

---

# LineNumbering

Use this command to turn the Line Numbering feature on or off, and to set numbering and positioning options. The Line Numbering feature is used to place numbers to the left of the lines on a page. The LineNumbering command is the equivalent of setting options in the Line Numbering [dialog box](#).

## **syntax**

LineNumbering (State:*Value Set Member*; Start:*Numeric Expression*; Increment:*Numeric Expression*; NumberingPosition:*Measurement Expression*; BlankLines:*Value Set Member*)

## **parameters**

This command has five parameters:

### **State:**

This parameter specifies whether Line Numbering will be continuous, will restart each page, or will be turned off. This parameter is required with the LineNumbering command. The State parameter is the equivalent of choosing one of the options in the Line Numbering pop-up list in the Line Numbering dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Off!	0
RestartEachPage!	1
Continuous!	2

### **Start:**

This numeric expression specifies the number from which line numbering will increment. This parameter should be an integer.

### **Increment:**

This parameter specifies the increment value between line numbers. This parameter is the equivalent of setting a number in the Number Every x Lines box (where x is a number) in the Line Numbering dialog box.

### **NumberingPosition:**

This measurement expression specifies how far from the left side of the page the line numbers should appear. This parameter is the equivalent of the Position from Left Edge box in the Line Numbering dialog box.

### **BlankLines:**

This parameter specifies whether lines with no text should be included in line numbering. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Skip!	0
Count!	1

[LineNumberingDlg](#)

---

## LineNumberingDlg

The LineNumberingDlg command displays the Line Numbering dialog box, which lets the user set line numbering options at the insertion point in the active document. This command is the equivalent of choosing Line from the Layout menu, then choosing Numbering.

### **syntax**

LineNumberingDlg ( )

LineNumbering



---

## LineSpacing

The LineSpacing command inserts a Line Spacing code at the insertion point. The Line Spacing code determines the spacing of lines following that code. This command is the equivalent of setting the Spacing in the Line Spacing dialog box.

### **syntax**

LineSpacing (Spacing:*Numeric Expression*)

### **parameters**

This command has one parameter:

#### **Spacing:**

This numeric expression specifies how much space will be between each line. In single-spaced text, the distance from baseline to baseline is one line height. When you specify a new number for line spacing, the current line height is multiplied by that number. This parameter should have a value from 0.5 to 160. For example, if, based on your current line height, you wanted double-spaced text, this parameter should be equal to 2.

LineSpacingDlg

---

## LineSpacingDlg

The LineSpacingDlg command displays the Line Spacing dialog box, which lets the user insert a Line Spacing code. If Auto Code Placement is on (which is the default), the code will be placed at the beginning of the current page. If Auto Code Placement is turned off, the code will be placed at the insertion point. This code determines the spacing of lines following that code. The Line Spacing command is the equivalent of choosing Line from the Layout menu, then choosing Spacing.

### **syntax**

LineSpacingDlg ( )

LineSpacing

---

## ListDefine

The ListDefine command places a List Definition code ([DefMark:List]) at the insertion point . The list will be created in this location when the document is generated. The type and appearance of the list depends on the ListType and NumberingStyle parameters.

This command is the equivalent of setting options in the Define List dialog box.

### **syntax**

ListDefine (ListType:*Value Set Member*; NumberingStyle:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **ListType:**

This parameter sets the contents of the list. The members of this value set correspond to the List pop-up list in the Define List dialog box:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

UserList1!	0
UserList2!	1
UserList3!	2
UserList4!	3
UserList5!	4
FigureCaptions!	5
TableCaptions!	6
TextBoxCaptions!	7
UserBoxCaptions!	8
EquationCaptions!	9

#### **NumberingStyle:**

This parameter sets how the list will appear on the printed page. The members of this value set correspond to the options in the Numbering Format pop-up list in the Define List dialog box:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

NoNumbering!	0
NumFollowsEntry!	1
FollowsInParens!	2
FlushRight!	3
FlushLeaders!	4

ListDefineDlg

---

## ListDefineDlg

The ListDefineDlg command displays the Define List dialog box, which lets the user specify the type of list and the numbering format for the list. When the user chooses OK, a List Definition code ([DefMark:List]) is placed at the insertion point in the active document. The ListDefineDlg command is the equivalent of choosing Define from the Tools menu, then choosing List.

### **syntax**

ListDefineDlg ( )

ListDefine

---

## ListMark

The ListMark command marks the current selection for the list type (such as userlist or captions) specified in the ListType parameter. This command is the equivalent of choosing an option from the Number pop-up list in the Mark List [dialog box](#).

### **syntax**

ListMark (ListType:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **ListType:**

The members of this value set correspond to the options in the List Number dialog box. This parameter specifies the type of list in which the current selection should be included. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

UserList1!	0
UserList2!	1
UserList3!	2
UserList4!	3
UserList5!	4
FigureCaptions!	5
TableCaptions!	6
TextBoxCaptions!	7
UserBoxCaptions!	8
EquationCaptions!	9

[ListDefine](#)

[Generate](#)

---

## ListMarkDlg

The ListMarkDlg command displays the Mark List dialog box, which lets the user specify for which list to mark the selected text. This command is the equivalent of choosing Mark Text from the Tools menu, then choosing List.

### **syntax**

ListMarkDlg ( )

ListMark

---

## MacroAssignDlg

The MacroAssignDlg command displays the Assign Macro to Menu dialog box, which lets the user add frequently used macros to the macro menu, as well as remove macros from the macro menu. The MacroAssignDlg command is the equivalent of choosing Assign to Menu from the Macro menu.

### **syntax**

MacroAssignDlg ( )

MacroMenuAppend

MacroMenuDelete

---

## MacroMenuAppend

The MacroMenuAppend command adds up to nine macro names to the Macro menu. The user can then choose these macro names from the Macro menu, and a corresponding macro plays. This command is similar to inserting macro names into the Macro menu by using the Assign Macro to Menu and Insert Macro Menu Item dialog boxes.

The Macro menu can contain a maximum of nine macros at a time. The MacroMenuAppend command cannot replace items currently on the Macro menu, so some parameters in the MacroMenuAppend command may be ignored. For example, suppose the Macro has six macros on it. If a MacroMenuAppend command with five MacroText and MacroName parameters is encountered, only the first three macros will be appended to the list. The other two macros will be ignored, since the total number of macros on the list is already at nine.

If you add macro descriptions to the Macro menu while recording a macro, a MacroMenuDelete ( ) command is first inserted into the macro. This causes all Macro menu items to be removed. This command is followed by the MacroMenuAppend command, which appends the specified macro at the top of the menu and reinserts all the macros you had previously added to the menu, specifying the description and pathname for every item in the final Macro menu.

### **syntax**

MacroMenuAppend (MenuText1:*Character Expression*; MacroName1:*Character Expression*; MenuText2:*Character Expression*; MacroName2:*Character Expression*; MenuText3:*Character Expression*; MacroName3:*Character Expression*; MenuText4:*Character Expression*; MacroName4:*Character Expression*; MenuText5:*Character Expression*; MacroName5:*Character Expression*; MenuText6:*Character Expression*; MacroName6:*Character Expression*; MenuText7:*Character Expression*; MacroName7:*Character Expression*; MenuText8:*Character Expression*; MacroName8:*Character Expression*; MenuText9:*Character Expression*; MacroName9:*Character Expression*)

### **parameters**

This command has 18 parameters:

**MenuText1: Mext2: MenuText3: MenuText4: MenuText5: MenuText6: MenuText7: MenuText8: MenuText9:**

These character expressions specify the text that will appear in the Macro menu. Each MenuTextx (where x is a number from 1 to 9) parameter corresponds to a MacroName<sub>x</sub> parameter.

The number in the MenuText parameters does not specify which menu item that text will occupy. The menu text will be placed in the first available position. For example, suppose a Macro menu already has four macro items. The text for MenuText1 would occupy the fifth space.

**MacroName1: MacroName2: MacroName3: MacroName4: MacroName5: MacroName6: MacroName7: MacroName8: MacroName9:**

These character expressions specify the pathname of the macros that correspond to the MenuText parameters.

MacroAssignDlg

MacroMenuDelete





---

## MacroMenuDelete

The MacroMenuDelete command removes the specified macro description from the Macro menu. It does not delete the macro itself.

### **syntax**

MacroMenuDelete (MacroNumber:*Numeric Expression*)

### **parameters**

This command has one parameter. If this parameter is omitted, all macro descriptions are removed from the Macro menu.

#### **MacroNumber:**

This numeric expression specifies which macro description to remove from the Macro menu. This numeric expression should have a value from 1 to 9, which corresponds to the order of the macro descriptions on the Macro menu.

[MacroAssignDlg](#)

[MacroMenuAppend](#)

---

## MacroStatusPrompt

The MacroStatusPrompt command lets the macro place (or remove) a message in the WordPerfect status bar. This message remains until it is turned off (using this command) or until the macro ends.

If a macro status prompt message is already displayed when the MacroStatusPrompt command is encountered, the new message replaces the old message.

### **syntax**

MacroStatusPrompt (State:*Value Set Member*; Prompt:*Character Expression*)

### **parameters**

This command has two parameters:

#### **State:**

This parameter specifies whether the macro status prompt message is turned on or off. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

#### **Prompt:**

This character expression specifies the message to appear on the status bar. This expression has a limit of 128 bytes. Each standard character is one byte in length; extended characters are longer. However, the actual number of characters that will fit in the status prompt depends on the width of the characters in the prompt and the size of the WordPerfect application window.

If the State parameter is set to Off!, the Prompt parameter is ignored and can be omitted.

### PROMPT

### Show Me!

---

## MailMAPIDlg

The MailMAPIDlg command invokes the e-mail dialog box that allows the user to send mail via any electronic mail package that conforms to the MAPI standard, such as Microsoft Mail. To make MAPIMail available, you must install a package that uses the MAPI standard and insert the following line in your WPC.INI file:

```
{Mail}  
MAPI=&Mail
```

The MailMAPIDlg command is the equivalent of choosing Mail from the File menu.

If you have the capabilities of using e-mail packages that conform to the VIM standard (cc:Mail) or if you have WordPerfect Mail for Windows, WordPerfect can send e-mail on those systems using the MailVIMDlg command or the MailWPDlg command. There is a priority for these systems, however, and only one mail package can be listed and available under the File menu at a time. WordPerfect Mail for Windows will be the only one listed for use if it is available with one or both of the other mail standards. If VIM and MAPI are the only mail standards possible, VIM will be the only one available for use. The priority is WPMail, VIMMail, MAPIMail. To make a mail of a lower priority available, you must delete the higher priority listings.

### **syntax**

MailMAPIDlg ( )

MailVIMDlg

MailWPDlg

---

## MailVIMDIg

The MailVIMDIg command invokes the e-mail dialog box that allows the user to send mail via any electronic mail package that conforms to the VIM standard, such as cc:Mail from Lotus. To make VIMMail available, you must install a package that uses the VIM standard and insert the following line in your WPC.INI file:

```
{Mail}  
VIM=&Mail
```

The MailVIMDIg command is the equivalent of choosing Mail from the File menu.

If you have the capabilities of using e-mail packages that conform to the VIM standard (cc:Mail from Lotus) or if you have WordPerfect Mail for Windows, WordPerfect can send e-mail on those systems using the MailMAPIDlg command or the MailWPDlg command. There is a priority for these systems, however, and only one mail package can be listed and available under the File menu at a time. WordPerfect Mail for Windows will be the only one listed for use if it is available with one or both of the other mail standards. If VIM and MAPI are the only mail standards possible, VIM will be the only one available for use. The priority is WPMail, VIMMail, MAPIMail. To make a mail of a lower priority available, you must delete the higher priority listings.

### **syntax**

MailVIMDIg ( )

MailMAPIDlg

MailWPDlg

---

## MailWPDlg

The MailWPDlg command invokes the e-mail dialog box that allows the user to send mail via WordPerfect's electronic mail package, WordPerfect Mail for Windows. Unless you have installed WordPerfect Mail for Windows, this feature and command will not be available. This command is the equivalent of choosing Mail from the File menu.

If you have the capabilities of using e-mail packages that conform to the VIM standard (cc:Mail from Lotus) or the MAPI standard (Microsoft Mail), WordPerfect can send e-mail on those systems using the MailVIMDlg command or the MailMAPIDlg command. There is a priority for these systems, however, and only one mail package can be listed and available under the File menu at a time. WordPerfect Mail for Windows will be the only one listed for use if it is available with one or both of the other mail standards. If VIM and MAPI are the only mail standards possible, VIM will be the only one available for use. The priority is WPMail, VIMMail, MAPIMail. To make a mail of a lower priority available, you must delete the higher priority listings.

### **syntax**

MailWPDlg ( )

MailMAPIDlg

MailVIMDlg

---

## MarginRelease

Depending on the location of the insertion point, the MarginRelease command performs different functions. When the insertion point is in a table or outline structure, this command moves the insertion point back to the previous cell or level. Otherwise, this command inserts a Margin Release code ([Mar Rel]), which moves the insertion point back to the previous tab stop. This command is the equivalent of pressing Shift+Tab.

If you want a command to insert a Margin Release code and move the insertion point back to the previous tab stop while in a table or outline structure, use the MarginReleaseInsert command.

### **syntax**

MarginRelease ( )

MarginReleaseInsert

---

## MarginReleaseInsert

The MarginReleaseInsert command inserts a Margin Release code ([MarRel]), which moves the insertion point back to the previous tab stop. This command is the equivalent of pressing Ctrl+Shift+Tab.

If the insertion point is in a table or outline structure, and you want a command to move the insertion point back to the previous cell or level without inserting a Margin Release code, use the MarginRelease command.

### **syntax**

MarginReleaseInsert ( )

MarginRelease



---

## MasterDocCondense

The MasterDocCondense command changes the expanded master document back to its condensed form, while optionally saving subdocuments. Macro execution is suspended until the master document has been condensed. This command is the equivalent of choosing the options and commands in the Condense Master Document [dialog box](#).

### **syntax**

MasterDocCondense (SaveSubdocuments:*Value Set Member*; ConfirmOverwrite:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **SaveSubdocuments:**

This parameter specifies whether changes to the subdocuments will be saved. It corresponds to choosing the Yes or No [button](#) in the Condense Master Document dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

#### **ConfirmOverwrite:**

This parameter specifies whether the user will be prompted any time a modified subdocument would overwrite an existing [file](#) by the same name. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

[MasterDocCondenseDlg](#)

---

## MasterDocCondenseDlg

The MasterDocCondenseDlg command displays the Condense Master Document dialog box, which lets the user change the expanded master document back to its condensed form, while optionally saving subdocuments. Macro execution is suspended until the master document has been condensed. This command is the equivalent of choosing Master Document from the Tools menu, then choosing Condense Master.

### **syntax**

MasterDocCondenseDlg ( )

MasterDocCondense

---

## MasterDocExpand

The MasterDocExpand command expands the active document (which should be a master document). The full text of each subdocument is retrieved into the master document. This command is the equivalent of choosing Master Document from the Tools menu, then choosing Expand Master.

### **syntax**

MasterDocExpand ( )

MasterDocCondense

---

## MasterDocSubdocDlg

The MasterDocSubdocDlg command displays the Include Subdocument dialog box, which lets the user specify the name of a subdocument to include in the master document currently being created. The MasterDocSubdocDlg command is the equivalent of choosing Master Document from the Tools menu, then choosing Subdocument.

### **syntax**

MasterDocSubdocDlg ( )

MasterDocSubdocInsert

---

## MasterDocSubdocInsert

The MasterDocSubdocInsert command inserts a subdocument mark at the insertion point in the master document. This command is the equivalent of specifying a document name and path in the Filename text box of the Include Subdocument dialog box.

### **syntax**

MasterDocSubdocInsert (SubDocumentName:*Character Expression*)

### **parameters**

This command has one parameter:

#### **SubDocumentName:**

This character expression is the name or full path of the subdocument to be included in the master document.

---

# MENU

The MENU command displays a menu with the options you specify. Each option in the menu has a corresponding letter or number (depending on the Mnemonic Type parameter) to its left. For instance, if you choose letters for the Mnemonic Type (set the parameter to LETTER), the first option is labeled A, the second is labeled B, and so forth. If you choose numbers for the Mnemonic Type (set the parameter to DIGIT), the first option is labeled 1, the second is labeled 2, and so forth. Mnemonic Type affects only how the options are displayed, not what is assigned to the variable when the user selects an option.

The user can select an option by clicking on it, or by typing the letter or number corresponding to that option. A numeric value is then assigned to the variable according to the option the user selects. For example, if the user selects the first option (regardless of whether the Mnemonic Type is a letter or a digit), "1" is assigned to the variable. If the user selects the third option, "3" is assigned to the variable, and so forth. If the user presses **Alt**, the menu is dismissed.

The MENU command can be used in conjunction with other commands, such as CASE and CASE CALL, to specify the course of action for the macro.

## **syntax**

MENU (*Variable; Mnemonic Type; Horizontal Position; Vertical Position; {Option; Option; . . . Option}*)

## **parameters**

This command has several parameters:

### **Variable**

Any variable is valid for this parameter.

### **Mnemonic Type**

This parameter specifies whether letters or numbers will correspond to the options in the menu. The values for this parameter are text, not character expressions, so these values should not be enclosed in quotation marks. The valid values for this parameter are DIGIT and LETTER.

### **Horizontal Position**

This numeric expression sets the horizontal position of the menu. The position is measured in pixels from the leftmost point of the active application window to the left side of the menu. Since different screen types have different numbers of pixels, where the menu displays depends on the resolution of your screen. If you omit this parameter, the menu will appear in the center of the screen.

### **Vertical Position**

This numeric expression sets the vertical position of the menu. The position is measured in pixels from the top of the active application window to the top of the menu. Since different screen types have different numbers of pixels, where the menu displays depends on the resolution of your screen. If you omit this parameter, the menu will appear in the center of the screen.

### **Option**

This parameter is a series of character expressions. The number of options is determined by whether you choose LETTER or DIGIT as the mnemonic type. If you use letters, you can have 26 options; if you use digits, you can have 9 options. Each option is a character expression and will be shown as an item in the menu.

### **Note**

Although you can display mnemonics in the options on your menu by placing an ampersand

(&) immediately before the letter you would like underlined on the menu, this will not affect which letter the user presses to select that option. For example, if your options are:

```
MENU (NAME; LETTER;;; {"&Business"; "&Personal";})
```

Your menu will display the mnemonics for the B and the P, but the mnemonics are not available for option selection; the user must still press the A or the B to select either option.

CASE

CASECALL

Show Me!

---

## MenuAddItem

The MenuAddItem command allows you to add macros as menu items to your WordPerfect menu. You can add items to the standard WordPerfect menu or you can create your own personalized submenu and add items to that.

A submenu is created by first adding a menu item to the main menu. Give the item an ItemName that will later be your submenu name. For example, if you wish to create a submenu called "Letters" that allows you to send letters to a large number of people, you would call your first ItemName, "Letters." This ItemName, "Letters," then becomes the MenuName for the next menu item you add. For example, you might have a submenu, "Letters," with menu items called, "Business"; "Family"; "Friends"; and "Other." Each menu item on the "Letters" submenu could bring up a template or a style that you had created for the letter type you wished to use.

### syntax

MenuAddItem (MenuName: *Character Expression*; ItemName: *Character Expression*; Offset: *Numeric Expression*; Type: *ValueSet Member*; ItemText: *Character Expression*; MacroName: *Character Expression*; Prompt: *Character Expression*; HelpFile: *Character Expression*; HelpId: *Numeric Expression*)

### parameters

This command has six parameters.

#### MenuName

This parameter is a character expression. The Menu Name is the name of the menu to which you are adding a menu item. If you wish to create a submenu, you must first add a menu item to the main menu, such as "Letters", then add subsequent items to the "Letters" menu item, making "Letters" your new MenuName.

#### ItemName

This parameter is a character expression. The ItemName is the name of the item you are adding to the menu. If you are creating submenus, the ItemName becomes your submenu name for subsequent menu items.

#### Offset

This parameter is a numeric expression. The Offset specifies the location of the menu item. If, for example, the Offset parameter is set to 1, then the new menu item will be in front of the first item on the menu specified in the MenuName parameter. If the Offset parameter is set to 10, the new menu item will be placed in front of the tenth item on the menu specified in the MenuName parameter. To place the new menu item at the absolute beginning of the menu, set the Offset parameter to 0. The highest number of menu items allowed is 255, although only about 20-30 show at any given time.

#### Type

This parameter is a value set member. The Type parameter designates what type of menu item you are creating: a macro, a separator, or a submenu. The valid values are:

Member	Numeric Equivalent
Macro!	0
Separator!	1
Submenu!	2

#### ItemText

This parameter is a character expression. The ItemText is the exact text that will appear on the menu for the menu item you have created.



**MacroName**

This parameter is a character expression. The MacroName is the name of the macro that will play when this menu item is chosen by the user. This parameter is optional.

**Prompt**

This parameter is a character expression. The Prompt is the long prompt, the text that appears on the status line at the bottom of the WordPerfect window, that will be displayed when this menu item is selected. This parameter is optional.

**HelpFile**

This parameter is a character expression. The HelpFile designates the Windows help file to be used if the user selects help for the menu item. This parameter is optional.

**HelpId**

This parameter is a numeric expression. The HelpId is the Id number that links to the help file so that when the user selects help for the menu item, the correct help file is displayed. This parameter is optional.

MenuDeleteItem

---

## MenuDeleteItem

The MenuDeleteItem command deletes a menu item that you have created from the menu you specify.

### **syntax**

MenuDeleteItem (MenuName: *Character Expression*; ItemName: *Character Expression*)

### **parameters**

This command has two parameters.

#### **MenuName**

This parameter is a character expression. The MenuName is the name of the menu or submenu from which you are deleting a menu item.

#### **ItemName**

This parameter is a character expression. The ItemName is the name of the menu item you are deleting.

[MenuAddItem](#)

---

## MergeCodesDlg

The MergeCodesDlg command displays the Insert Merge Codes dialog box, which lets the user insert merge codes into the active document. This command is the equivalent of choosing Merge from the Tools menu, then choosing Merge Codes.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

MergeCodesDlg ( )

MergeInsertCode

---

## MergeConvert

The MergeConvert command converts WordPerfect 4.2 or 5.0 merge codes in the active document to the new merge format. This command is the equivalent of choosing Merge from the Tools menu, choosing Convert, then choosing OK in the Convert Old Merge Codes dialog box.

### **syntax**

MergeConvert ( )

MergeConvertMsgDlg

---

## MergeConvertMsgDlg

The MergeConvertMsgDlg command displays the Convert Old Merge Codes message box, which lets the user specify whether to convert any WordPerfect 4.2 or 5.0 merge codes to the new format. This command is the equivalent of choosing Merge from the Tools menu, then choosing Convert.

When WordPerfect encounters the MergeConvertMsgDlg command, the macro pauses until the user chooses OK or Cancel. Depending on what the user chooses, WordPerfect either converts old merge codes and then continues the macro, or just continues the macro.

### **syntax**

MergeConvertMsgDlg ( )

MergeConvert

---

## MergeDOSText

The MergeDOSText command specifies the field and record delimiters for a DOS delimited file that will be used as a secondary file in a merge. This command should be executed before a MergeExecute command when the secondary file is a DOS text file. The MergeDOSText command is the equivalent of setting the values in the Text File Delimiters dialog box.

### **syntax**

MergeDOSText (FieldBegin:*Character Expression*; FieldEnd:*Character Expression*;  
RecordBegin:*Character Expression*; RecordEnd:*Character Expression*)

### **parameters**

This command has four parameters:

#### **FieldBegin: FieldEnd: RecordBegin: RecordEnd:**

These character expressions specify the text and/or codes used to delimit (in other words, begin and/or end) the fields and records in the text file. For example, your text file may begin and end each field with quotes and end each record with a carriage return. Only those parameters that apply to your text file need to be filled in.

If the delimiters of your text file are characters such as quotes or commas, simply type the character in the appropriate parameter. If the delimiters of your text file are codes such as tabs, line feeds, form feeds, or carriage returns, specify those codes in the Text File Delimiters dialog box while recording a macro. This inserts an "embedded" code into the character string, which can be viewed using [Reveal Codes](#) when you edit the macro. If you want to insert an embedded code into an existing FieldBegin, FieldEnd, RecordBegin, or RecordEnd parameter in a macro, record another macro containing that code, open that macro, copy the embedded code onto the [Clipboard](#), then paste it into that parameter.

[MergeExecute](#)

[PrefMergeDelimit](#)

---

## MergeEndField

The MergeEndField command inserts an {END FIELD} merge code at the insertion point in the active document. This command is the equivalent of choosing Merge from the Tools menu, then choosing End Field.

### **syntax**

MergeEndField ( )

MergeCodesDlg

MergeInsertCode

---

## MergeEndRecord

The MergeEndRecord command inserts an End Record merge code ({END RECORD}) at the insertion point in the active document. This command is the equivalent of choosing Merge from the Tools menu, then choosing End Record.

### **syntax**

MergeEndRecord ( )

MergeCodesDlg

MergeInsertCode



---

## MergeExecute

The MergeExecute command begins the merging of two specified documents. This command is the equivalent of specifying a primary and secondary file in the Merge dialog box.

### **syntax**

MergeExecute (PrimaryFile:*Character Expression*; SecondaryFile:*Character Expression*)

### **parameters**

This command has two parameters:

#### **PrimaryFile:**

This character expression is the name or full pathname of the primary file to be used in the merged document.

#### **SecondaryFile:**

This character expression is the name or full pathname of the secondary file to be used in the merged document. This parameter is optional.

MergeFilesDlg

---

## MergeFieldDlg

The MergeFieldDlg command displays the Insert Merge Code dialog box, which lets the user specify a field name or number. This command is the equivalent of choosing Merge from the Tools menu, then choosing Field.

### **syntax**

MergeFieldDlg ( )

---

## MergeFilesDlg

The MergeFilesDlg command displays the Merge dialog box, which lets the user begin merging two specified documents. This command is the equivalent of choosing Merge from the Tools menu, then choosing Merge.

### **syntax**

MergeFilesDlg ( )

MergeExecute

---

## MergeInputDialog

The MergeInputDialog command displays the Insert Merge Code dialog box, which lets the user specify an input message to be used in a Merge primary file. This command is the equivalent of choosing Merge from the Tools menu, then choosing Input.

### **syntax**

MergeInputDialog ( )

---

## MergeInsertCode

The MergeInsertCode command inserts a specified Merge code, as well as any specified parameters.

This command is similar to choosing a command in the Insert Merge Codes dialog box, but this command does not prompt you for the parameters used with any of the merge commands. The text and tildes of the merge commands parameters must be included in the ParameterString parameter of the MergeInsertCode command.

### **syntax**

MergeInsertCode (MergeCode:*Value Set Member*; ParameterString:*Character Expression*)

### **parameters**

This command has two parameters:

#### **MergeCode:**

This parameter specifies the merge code to be inserted at the insertion point. This value set corresponds to the options in the Insert Merge Codes dialog box.

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Assign!	0
Bell!	1
Break!	2
Call!	3
CancelOff!	4
CancelOn!	5
Case!	6
CaseCall!	7
ChainMacro!	8
ChainPrimary!	9
ChainSecondary!	10
Char!	11
Comment!	12
Cton!	13
Date!	14
Document!	15
Else!	16
EndField!	17
EndFor!	18
EndIf!	19
EndRecord!	20
EndWhile!	21
Field!	22

For!	23
Go!	25
If!	26
IfBlank!	27
IfExists!	28
IfNotBlank!	29
Keyboard!	30
Label!	31
Local!	32
Look!	33
Mid!	34
MergeCommand!	35
NestMacro!	36
NestPrimary!	37
NestSecondary!	38
Next!	39
NextRecord!	40
Ntoc!	41
Process!	42
OnCancel!	43
OnError!	44
PageOff!	45
PageOn!	46
Print!	47
Prompt!	48
Quit!	49
Return!	50
ReturnCancel!	51
ReturnError!	52
Rewrite!	53
StepOff!	54
StepOn!	55
SubstPrimary!	56
SubstSecondary!	57
System!	58
Text!	59
Variable!	60

Wait!	61
While!	62
StatusPrompt!	63
Input!	64
Len!	65
FieldNames!	66
Stop!	67
Otherwise!	68

**ParameterString:**

This character expression sets the text and tildes (delimiters) for the merge command specified in the MergeCode parameter. This command may be excluded if the merge code doesn't use any parameters.

MergeCodesDlg

---

## MergeNextRecord

The MergeNextRecord command inserts a Next Record code ({NEXT RECORD}) at the insertion point in the active document. This command is the equivalent of choosing Merge from the Tools menu, then choosing Next Record.

### **syntax**

MergeNextRecord ( )



---

## MergePageOff

The MergePageOff command inserts a Page Off merge code ({PAGE OFF}) at the insertion point in the active document. During a merge, the Page Off command eliminates the hard page between copies of the primary file in the merged document. The Page On command ({PAGE ON}) counteracts the Page Off command, and can be inserted using the MergeInsertCode macro command. This command is the equivalent of choosing Merge from the Tools menu, then choosing Page Off.

### **syntax**

MergePageOff ( )

MergeInsertCode

---

## MergeVariableGet

The MergeVariableGet command assigns the contents of a specified merge global variable to a macro variable. This command is useful for using merge information in a macro, since merge variables and macro variables are not interchangeable.

If you want to assign macro information to a merge variable, use the MergeVariableSet command.

### **syntax**

MergeVariableGet (MacroVariableName:*Variable*; MergeVariable:*Character Expression*)

### **parameters**

This command has two parameters:

#### **MacroVariableName:**

The contents of the merge variable will be assigned to this variable.

#### **MergeVariable:**

This character expression specifies the name of the merge variable, the contents of which will be assigned to a macro variable.

[MergeVariableSet](#)

[Show Me!](#)

---

## MergeVariableSet

The MergeVariableSet command puts the value of a specified character expression into a merge variable. Merges then access the information in that character expression through the variable name specified in the MergeVariable parameter. This command, along with the MergeVariableGet command, allows programmable merges and macros to share information.

### **syntax**

MergeVariableSet (MergeVariable:*Character Expression*; StringValue:*Character Expression*)

### **parameters**

This command has two parameters:

#### **MergeVariable:**

This character expression specifies the name of the merge variable that will contain the StringValue parameter.

#### **StringValue:**

This character expression specifies the information to be assigned to the merge variable. Keep in mind that this is a character expression, so any numeric values you want to pass into a merge variable must first be converted to text by using the NUMSTR programming command.

Also, macro variables can potentially hold much more information than merge variables can. If the size of the macro character expression is greater than the merge variable can handle, the character expression will be truncated.

[Show Me!](#)

---

## NEWDEFAULT

The NEWDEFAULT command lets you specify a new product prefix as the default. After the macro encounters a NEWDEFAULT command, any product commands without a product prefix will be treated as if they have the product prefix specified in the Product Prefix parameter of the NEWDEFAULT command.

Only one product prefix may be the default at any point in time. Several NEWDEFAULT commands may be included in a single macro, but each NEWDEFAULT command applies only until either the next NEWDEFAULT command or an ENDAPP command specifying a product prefix is encountered.

This command assumes that a product prefix is set as the default when the macro executes. If no product prefix is currently set as the default, a compilation error will result when you try to play the macro.

When you define a product prefix in the APPLICATION command, you can also set it as the default application.

The NEWDEFAULT command is executed differently than most commands; it is implemented when you compile the macro, rather than when you run the macro. This means that the NEWDEFAULT command is always effective from its location in the macro, even if the NEWDEFAULT command itself is not encountered when the macro runs. For example, if a NEWDEFAULT command is located within an IF-ENDIF conditional statement, the NEWDEFAULT command is effective from its location forward, regardless of whether the IF argument is true.

### **syntax**

NEWDEFAULT (*Product Prefix*)

### **parameters**

This command has one parameter:

#### **Product Prefix**

This parameter is two letters. The Product Prefix parameter must correspond to a product prefix set in an APPLICATION command earlier in the macro. This parameter is not a character expression, so it should not be enclosed in quotation marks.

### APPLICATION

---

## NUMSTR

The NUMSTR command converts a number or measurement to its text equivalent. For instance, the number 123 would convert to the text string "123". Although the two look the same on your screen, they are understood differently by the computer. You cannot add a number or measurement to a character expression. Similarly, you cannot add text to a number or measurement expression. This command, along with the STRNUM command, allows you to change one type of value to the other.

The NUMSTR command finds the expression parameter's character equivalent and assigns it to the variable specified in the Variable parameter. The Decimal Places parameter tells the computer how many digits beyond the decimal to include in the variable. The decimal point recognized by this command is that defined by the "sDecimal" setting in the "[intl]" section of the Windows WIN.INI [file](#). If the expression is longer than the specified decimal places, NUMSTR rounds the expression up. For instance, if a macro contains the command NUMSTR (Var01; 3; 7654.3219), Var01 would contain the character expression 7654.322.

### **syntax**

NUMSTR (*Variable; Decimal Places; Expression*)

### **parameters**

This command has three parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Decimal Places**

Decimal Places is a numeric expression. This parameter specifies how many places beyond the decimal will be included in the character expression. This parameter should be an integer with a value from 0 to 16.

If you omit the Decimal Places parameter, this command will convert six places beyond the decimal to the character expression.

#### **Expression**

This parameter is a numeric or measurement expression. This is the expression that is to be converted to its character expression equivalent.

### STRNUM

### Show Me!

---

## NoteNext

Use the NoteNext command to open the next subdocument of the type that is currently active. For instance, if the Endnote 2 subdocument is currently active, this command causes the Endnote 3 subdocument to become active. The NoteNext command works with endnotes, footnotes, headers, and footers. This command is the equivalent of choosing the Next button in a Footnote or Endnote subdocument window.

If no upcoming subdocuments of the active type exist when this command is encountered, an error occurs.

### **syntax**

NoteNext ( )

NotePrevious

---

## NoteNumber

The NoteNumber command inserts the current footnote or endnote number into the text editor at the insertion point. If a text editor is not active, this command is ignored. The NoteNumber command is the equivalent of choosing Note Number in the footnote or endnote editor.

### **syntax**

NoteNumber ( )

---

## NotePrevious

Use the NotePrevious command to open the preceding subdocument of the type that is currently active. For instance, if the Endnote 2 subdocument is currently active, this command causes the Endnote 1 subdocument to become active. The NotePrevious command works with endnotes, footnotes, headers, and footers. This command is the equivalent of choosing the Previous button in a Footnote or Endnote subdocument window.

If no preceding subdocuments of the active type exist when this command is encountered, an error occurs.

### **syntax**

NotePrevious ( )

NoteNext



---

## NumberFormat

The NumberFormat command sets the characters to be used for decimal alignment and as the thousands separator. This command is the equivalent of specifying a Decimal Align Character and Thousands Separator in the Insert Special Codes [dialog box](#).

### **syntax**

NumberFormat (ThousandsSeparator:*Character Expression*; DecimalAlignChar:*Character Expression*)

### **parameters**

This command has two parameters:

#### **ThousandsSeparator:**

This character expression specifies the character that will separate groups of thousands in WordPerfect. The ThousandsSeparator parameter should be only one character long.

#### **DecimalAlignChar:**

This character expression specifies where text is aligned on decimal tabs. This parameter should be only one character long.

---

# ONCANCEL

This command sends macro execution to a specified LABEL command when a cancel is encountered. If a cancel occurs before an ONCANCEL command is encountered, the macro stops. If a cancel occurs after an ONCANCEL command, macro execution goes to the LABEL command specified in the Label parameter. If a cancel occurs after two or more ONCANCEL commands, the macro goes to the LABEL command most recently specified in an ONCANCEL command in that level of execution.

ONCANCEL is similar to the ONCANCEL CALL command. However, when you use the ONCANCEL CALL command to branch to a specified LABEL command in the macro, you include a RETURN command later in the macro. This returns macro execution to the command which immediately follows the command most recently encountered when the cancel condition occurred.

## **syntax**

ONCANCEL (*Label*@)

## **parameters**

This command has one parameter:

### **Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

[ASSERTCANCEL](#)

[ONCANCEL CALL](#)

[Show Me!](#)

---

## ONCANCEL CALL

The ONCANCEL CALL command branches to the specified LABEL command when a cancel is encountered during macro execution. The macro runs from that point until it encounters a RETURN command. Macro execution is then returned to the command immediately following the command most recently executed when the cancel was encountered.

The ONCANCEL CALL command is similar to the ONCANCEL command. ONCANCEL, however, simply transfers macro execution to the specified LABEL command without expecting a RETURN.

If an ONCANCEL or ONCANCEL CALL command has not been encountered in a macro when a cancel is encountered, the macro stops.

### **syntax**

ONCANCEL CALL (*Label*@)

### **parameters**

This command has one parameter:

#### **Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

[ASSERTCANCEL](#)

[ONCANCEL](#)

[RETURN](#)

[Show Me!](#)

---

## **ONERROR**

The ONERROR command sends macro execution to the specified LABEL command when an error occurs.

ONERROR is similar to the ONERROR CALL command. However, when the ONERROR CALL command transfers macro execution to a LABEL command in the macro, a RETURN command should be included to return the macro execution to the command immediately following the command that triggered the error condition.

If an ONERROR or ONERROR CALL command has not been encountered in a macro when an error is encountered, the macro stops. If an error occurs after two or more ONERROR commands, the macro goes to the LABEL command most recently specified in an ONERROR command in the active macro.

### **syntax**

ONERROR (*Label*@)

### **parameters**

This command has one parameter:

#### **Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

ASSERTERROR

ONERROR CALL

Show Me!

---

## ONERROR CALL

The ONERROR CALL command branches to the specified LABEL command when an error is encountered during macro execution. The macro runs from that point until it encounters a RETURN command. Macro execution is then returned to the command immediately following the command most recently executed when the error was encountered.

The ONERROR CALL command is similar to the ONERROR command. ONERROR, however, simply transfers macro execution to the specified LABEL command without expecting a RETURN.

If an ONERROR or ONERROR CALL command has not been encountered in a macro when an error occurs, the macro stops.

### **syntax**

ONERROR CALL (*Label*@)

### **parameters**

This command has one parameter:

#### **Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

ASSERTERROR

ONERROR

RETURN

Show Me!

---

## ONNOTFOUND

The ONNOTFOUND command sends macro execution to a specified LABEL command when a "not found" state is encountered. What constitutes a not found state depends on the application. Macro execution is transferred to the LABEL command most recently specified in an ONNOTFOUND command in that level of execution.

ONNOTFOUND is similar to the ONNOTFOUND CALL command. However, when the ONNOTFOUND CALL command transfers macro execution to a LABEL command in the macro, a RETURN command should be included to return macro execution to the command immediately following the command most recently executed when the "not found" occurred.

If an ONNOTFOUND or ONNOTFOUND CALL command has not been encountered in a macro when a not found state is encountered, the macro stops.

### **syntax**

ONNOTFOUND (*Label*@)

### **parameters**

This command has one parameter:

#### **Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

[ASSERTNOTFOUND](#)

[ONNOTFOUND CALL](#)

[Show Me!](#)

---

## ONNOTFOUND CALL

The ONNOTFOUND CALL command branches to the specified LABEL command when a search fails during macro execution. The macro runs from that point until it encounters a RETURN command. Macro execution is then returned to the command immediately following the command where the macro was when the search failed.

The ONNOTFOUND CALL command is similar to the ONNOTFOUND command. ONNOTFOUND, however, simply transfers macro execution to the specified LABEL command without expecting a RETURN.

If an ONNOTFOUND or ONNOTFOUND CALL command has not been encountered in a macro when a not found state is encountered, the macro stops.

### **syntax**

ONNOTFOUND CALL (*Label@*)

### **parameters**

This command has one parameter:

#### **Label**

The label specified in this parameter must correspond to a LABEL command within the macro.

ASSERTNOTFOUND

ONNOTFOUND

RETURN

Show Me!

---

## OleInsertObject

The OleInsertObject command inserts an Ole object into the document. It is the equivalent of choosing Insert Object from the Edit menu and specifying in the dialog box the object type you are inserting.

### **syntax**

OleInsertObject (ObjectType: *Character Expression*)

### **parameter**

This command has one parameter.

### **ObjectType**

This parameter is a character expression. The ObjectType is a name that is available when you have an object in your document and the insertion point is positioned over the object. At that point, the ObjectName is displayed on the status bar at the bottom of the WordPerfect document window.

OleInsertObjectDlg



---

## OleInsertObjectDlg

The OleInsertObjectDlg command brings up the OleInsertObject dialog box. This allows the user to insert an Ole object into the WordPerfect document. It is the equivalent of choosing Insert Object from the Edit menu.

### **syntax**

OleInsertObjectDlg ( )

OleInsertObject

---

## OleLinksDlg

The OleLinksDlg command brings up the OleLinks dialog box. This allows the user to view, update, change, or cancel Ole links in the document. This command is the equivalent of choosing Links from Insert Object in the Edit menu.

### **syntax**

OleLinksDlg ( )

OleInsertObjectDlg

---

## OutlineCopy

The OutlineCopy command copies the outline family at the insertion point into memory. An "outline family" is the paragraph number and text on the line where the insertion point is located, plus any paragraph numbers and text of subordinate levels.

After the outline family has been copied, subsequent macro commands (either PosCharNext, PosCharPrevious, PosLineDown, or PosLineUp) can move the insertion point to the position where the outline family should be placed. A HardReturn command will then place a copy of that outline family at the insertion point.

This command is the equivalent of choosing Outline from the Tools menu, then choosing Copy Family.

### **syntax**

OutlineCopy ( )

OutlineDelete

OutlineMove

---

## OutlineDefine

The OutlineDefine command sets the options for an outline definition. This command is the equivalent of setting options in the Define Paragraph Numbering [dialog box](#).

### syntax

OutlineDefine (Style:*Character Expression*; InsertNumber:*Value Set Member*;  
AutoAdjust:*Value Set Member*; Level1:*Character Expression*; Level2:*Character Expression*;  
AttachLevel1:*Value Set Member*; Level3:*Character Expression*; AttachLevel2:*Value Set Member*;  
Level4:*Character Expression*; AttachLevel3:*Value Set Member*; Level5:*Character Expression*;  
AttachLevel4:*Value Set Member*; Level6:*Character Expression*;  
AttachLevel5:*Value Set Member*; Level7:*Character Expression*; AttachLevel6:*Value Set Member*;  
Level8:*Character Expression*; AttachLevel7:*Value Set Member*; StartingNumber:*Character Expression*)

### parameters

This command has 19 parameters:

#### Style:

This character expression specifies the style name to be used with the outline definition. This parameter is the equivalent of selecting a style in the Outline Styles dialog box.

#### InsertNumber:

This parameter specifies whether pressing Enter should insert a new paragraph number. This command is the equivalent of choosing Enter Inserts Paragraph Number from the Define Paragraph Numbering dialog box. The valid values are:

Member	Numeric Equivalent
--------	--------------------

---

No!	0
Yes!	1

#### AutoAdjust:

This parameter specifies whether WordPerfect should insert each new number at the level of the last number that was inserted. For example, if the last number was a second level number (such as A., B., C.), the next number will also be a second level number. The valid values are:

Member	Numeric Equivalent
--------	--------------------

---

No!	0
Yes!	1

#### Level1:

This character expression specifies how level 1 numbers will appear. This expression should contain characters that make reasonable level numbers and punctuation, such as I., A., 1., and so on. If you do not include the punctuation, it will not appear in your outline.

#### Level2:

This character expression specifies how level 2 numbers will appear. This expression should contain characters that make reasonable level numbers and punctuation, such as I., A., 1., and so on.

#### AttachLevel1:

This parameter specifies whether the level 1 paragraph number will be attached to the

beginning of level 2. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

### **Level3:**

This character expression specifies how level 3 numbers will appear. This expression should contain characters that make reasonable level numbers and punctuation, such as I., A., 1., and so on.

### **AttachLevel2:**

This parameter specifies whether the level 2 paragraph number will be attached to the beginning of level 3. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

### **Level4:**

This character expression specifies how level 4 numbers will appear. This expression should contain characters that make reasonable level numbers and punctuation, such as I., A., 1., and so on.

### **AttachLevel3:**

This parameter specifies whether the level 3 paragraph number will be attached to the beginning of level 4. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

### **Level5:**

This character expression specifies how level 5 numbers will appear. This expression should contain characters that make reasonable level numbers and punctuation, such as I., A., 1., and so on.

### **AttachLevel4:**

This parameter specifies whether the level 4 paragraph number will be attached to the beginning of level 5. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

### **Level6:**

This character expression specifies how level 6 numbers will appear. This expression should contain characters that make reasonable level numbers and punctuation, such as I., A., 1., and so on.

### **AttachLevel5:**

This parameter specifies whether the level 5 paragraph number will be attached to the beginning of level 6. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

**Level7:**

This character expression specifies how level 7 numbers will appear. This expression should contain characters that make reasonable level numbers and punctuation, such as I., A., 1., and so on.

**AttachLevel6:**

This parameter specifies whether the level 6 paragraph number will be attached to the beginning of level 7. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

**Level8:**

This character expression specifies how level 8 numbers will appear. This expression should contain characters that make reasonable level numbers and punctuation, such as I., A., 1., and so on.

**AttachLevel7:**

This parameter specifies whether the level 7 paragraph number will be attached to the beginning of level 8. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

**StartingNumber:**

This character expression lets the user change the starting paragraph number.

OutlineDelete

OutlineMove

OutlineOn

---

## OutlineDelete

The OutlineDelete command deletes the outline family at the insertion point. An "outline family" is the paragraph number and text on the line containing the insertion point, plus any paragraph numbers and text on subordinate levels. This command is the equivalent of choosing Outline from the Tools menu, then choosing Delete Family.

### **syntax**

OutlineDelete ( )

OutlineCopy

OutlineMove

---

## OutlineMove

The OutlineMove command moves the outline family at the insertion point into memory. An "outline family" is the paragraph number and text on the line containing the insertion point, plus any paragraph numbers and text on subordinate levels. The outline family is removed from its current location, so it can be moved to a new position. Subsequent macro commands (either PosCharNext, PosCharPrevious, PosLineDown, or PosLineUp) can move the insertion point to the position where the outline family should be placed.

A HardReturn command will then place a copy of that outline family at the insertion point. This command is the equivalent of choosing Outline from the Tools menu, then choosing Move Family.

### **syntax**

OutlineMove ( )

OutlineCopy

OutlineDelete



---

## **OutlineOff**

The OutlineOff command turns the Outline feature off at the insertion point. This command is the equivalent of choosing Outline from the Tools menu, then choosing Outline Off.

### **syntax**

OutlineOff ( )

OutlineOn

---

## OutlineOn

The OutlineOn command turns the Outline feature on at the insertion point. This command is the equivalent of choosing Outline from the Tools menu, then choosing Outline On.

### **syntax**

OutlineOn ( )

OutlineOff

---

## OverstrikeCreate

Use the OverstrikeCreate command to insert an overstrike character at the insertion point in the active document. This command is the equivalent of entering a string of characters in the Create/Edit dialog box after choosing Overstrike from the Font menu.

### **syntax**

OverstrikeCreate (Chars:*Character Expression*)

### **parameters**

This command has one parameter:

#### **Chars:**

This character expression specifies the characters used in the overstrike character.

OverstrikeEdit

OverstrikeCreateDlg

---

## OverstrikeCreateDlg

The OverstrikeCreateDlg command displays the Create/Edit Overstrike dialog box, which lets the user create an overstrike character to be inserted at the insertion point. This command is the equivalent of choosing Overstrike from the Font menu, then choosing Create.

### **syntax**

OverstrikeCreateDlg ( )

OverstrikeCreate

---

## OverstrikeEdit

Use the OverstrikeEdit command to edit a previously created overstrike character. WordPerfect determines which overstrike to edit by searching backward from the insertion point. If WordPerfect can't find an overstrike while searching backward, it searches forward from the beginning of the document for an overstrike character. WordPerfect changes the first overstrike it encounters to the overstrike specified in the Chars parameter. This command is the equivalent of entering a string of characters in the Create/Edit dialog box.

### **syntax**

OverstrikeEdit (Chars:*Character Expression*)

### **parameters**

This command has one parameter:

#### **Chars:**

This character expression specifies the characters used in the overstrike character.

OverstrikeCreate

OverstrikeEditDlg

---

## OverstrikeEditDlg

The OverstrikeEditDlg command displays the Create/Edit Overstrike dialog box, which lets the user edit an existing overstrike character. WordPerfect determines which overstrike to edit by searching backward from the insertion point. If WordPerfect can't find an overstrike while searching backward, it searches forward from the beginning of the document for an overstrike character.

WordPerfect lets the user edit the first overstrike it encounters. This command is the equivalent of choosing Overstrike from the Font menu, then choosing Edit.

### **syntax**

OverstrikeEditDlg ( )

OverstrikeEdit

---

## PAUSE

The PAUSE command temporarily stops the macro, allowing the user to use the computer as if there were no macro running. When the user chooses Pause from the Macro menu, the macro continues from the command below the PAUSE command. If a PROMPT message box is displayed when the PAUSE command is executed, the user also can resume the macro by choosing OK in the PROMPT message box.

You can determine whether a macro is paused in WordPerfect by looking at the Pause option in the Macro menu. If it is checked, the macro is paused.

If you want a macro to pause until a certain key is pressed, use the pausekey product command (see PauseKey).

### **syntax**

PAUSE

PauseKey

Show Me!

---

## PROMPT

The `PROMPT` command displays a message box with the title specified in the Title parameter, and the message specified in the Prompt parameter. The macro then continues normally. This message box will not remain on screen unless the PAUSE command is also used. If the PAUSE command is used in conjunction with the PROMPT and ENDPROMPT commands, the message box will stay on screen until the user chooses OK or Cancel. Only one PROMPT message box can be on the screen at a time. If a PROMPT message box is on the screen when another PROMPT command is encountered, the second PROMPT replaces the first.

If the user chooses Cancel in the message box, the macro behaves as if a cancel has occurred. For instance, if the macro has an ONCANCEL command, macro execution will go to the specified label. If the macro has no cancel handling built in, the macro will stop.

If the macro is paused when the user chooses OK in the PROMPT message box, the message box is cleared, and the macro resumes.

### **syntax**

`PROMPT (Title; Prompt; Icon; Horizontal Position; Vertical Position)`

### **parameters**

This command has five parameters:

#### **Title**

This character expression specifies the text to appear in the title bar of the dialog box.

#### **Prompt**

This character expression specifies the text that will appear in the dialog box.

#### **Icon**

This numeric expression lets you put an icon in the PROMPT message box to specify the type of information the message box contains.

If the numeric expression equals 1, the message box contains a stop sign icon.

If the numeric expression equals 2, the message box contains a question mark icon.

If the numeric expression equals 3, the message box contains an exclamation point icon.

If the numeric expression is equal to or greater than 4 or less than 0, the message box contains an information icon.

If this parameter is omitted or set to 0, the message box contains no icon. If you omit this parameter, you must still place a semicolon where the parameter would be.

#### **Horizontal Position**

This numeric expression sets the horizontal position of the PROMPT message box. The position is measured in pixels (dots on the screen) from the leftmost point of the active application window to the left side of the message box. If you omit this parameter, the message box will appear in the center of the screen. You must still place a semicolon where the parameter would be.

#### **Vertical Position**

This numeric expression sets the vertical position of the PROMPT message box. The position is measured in pixels (dots on the screen) from the top of the active application window to the top of the message box. If you omit this parameter, the message box will appear in the center of the screen. You must still place a semicolon where the parameter would be.



ENDPROMPT

MacroStatusPrompt

Show Me!

---

## PageCenter

The PageCenter command specifies whether the current page in the active document should be centered between the top and bottom margins. If the Set parameter is set to On!, a Center Page Code ([Center Pg]) is placed at the top of the current page. This command is the equivalent of choosing Page from the Layout menu, then choosing Center Page.

### **syntax**

PageCenter (Set:*Value Set Member*)

### **parameters**

This command has one parameter. The PageCenter command acts as a "toggle" if the parameter is left out.

### **Set:**

This parameter specifies whether the Center Page feature will be turned on or off for the current page. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
On!	1

---

## PageMargins

The PageMargins command sets the top, bottom, left, and right margins from the insertion point forward in the active document. This command is the equivalent of setting the margin measurements in the Margins dialog box.

### **syntax**

PageMargins (Left:*Measurement Expression*; Right:*Measurement Expression*;  
Top:*Measurement Expression*; Bottom:*Measurement Expression*)

### **parameters**

This command has four parameters:

#### **Left:**

This measurement expression sets the distance from the left side of the page to the left margin.

#### **Right:**

This measurement expression sets the distance from the right side of the page to the right margin.

#### **Top:**

This measurement expression sets the distance from the top of the page to the top margin.

#### **Bottom:**

This measurement expression sets the distance from the bottom of the page to the bottom margin.

PageMarginsDlg

---

## PageMarginsDlg

The PageMarginsDlg command displays the Margins dialog box, which lets the user specify top, bottom, left, and right margins. These margins will apply from the insertion point forward in the active document. The PageMarginsDlg command is the equivalent of choosing Margins from the Layout menu.

### **syntax**

PageMarginsDlg ( )

PageMargins

---

# PageNumbering

Use the PageNumbering command to set the position of page numbers, the type of characters to use with page numbers, a new page number, or to turn page numbering off. This command is the equivalent of setting options in the Page Numbering [dialog box](#).

## **syntax**

PageNumbering (Where:*Value Set Member*; Text:*Character Expression*; NewNumber:*Numeric Expression*; Style:*Value Set Member*)

## **parameters**

This command has four parameters:

### **Where:**

This parameter specifies where on each page the page number will be located. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

None!	0
TopLeft!	1
TopCenter!	2
TopRight!	3
TopAlternate!	4
BottomLeft!	5
BottomCenter!	6
BottomRight!	7
BottomAlternate!	8

### **Text:**

This character expression lets you make text a part of the page number. For instance, you may want your name or the title of the document to precede the page number.

### **NewNumber:**

This numeric expression sets the number with which to begin page numbering. This parameter may be omitted if you do not want to begin renumbering.

### **Style:**

This parameter specifies whether page numbers will appear as roman uppercase, roman lowercase, or arabic numerals. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Arabic!	0
LowerCaseRoman!	1
UpperCaseRoman!	2

[PageNumberingDlg](#)

[PageNumberInsert](#)



---

## PageNumberingDlg

The PageNumberingDlg command displays the Page Numbering dialog box, which lets the user specify where page numbers should appear, the type of page numbers (roman uppercase, roman lowercase, or arabic numerals), a new page number, and so forth. This command is the equivalent of choosing Page from the Layout menu, then choosing Numbering.

### **syntax**

PageNumberingDlg ( )

PageNumbering

PageNumberInsert

---

## PageNumberInsert

The PageNumberInsert command inserts the current page number at the insertion point. The PageNumberInsert command is the equivalent of choosing the Insert Page Number button in the Page Numbering dialog box.

If the user records a macro using the Page Numbering dialog box, the macro facility will insert PageNumberInsert (WithStyle:Yes!). This causes the current page number (^B) and any text in the Accompanying Text text box to be inserted at the insertion point.

If the user records a macro using Ctrl+Shift+P, the macro facility will insert PageNumberInsert ( ). This causes only the current page number (^B) to be inserted at the insertion point.

### **syntax**

PageNumberInsert (WithStyle:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **WithStyle:**

This parameter specifies whether the text in the Accompanying Text text box will be included with the current page number (^B). If this parameter is missing, the default is No!. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

PageNumberingDlg



---

## PageSuppress

Use the PageSuppress command to prevent specified headers, footers, and page numbers from printing on the current page. This command is the equivalent of setting options in the Suppress dialog box, then choosing OK.

### **syntax**

PageSuppress (HeaderA:*Value Set Member*; HeaderB:*Value Set Member*; FooterA:*Value Set Member*; FooterB:*Value Set Member*; PageNumber:*Value Set Member*; PageToBottom:*Value Set Member*)

### **parameters**

This command has six parameters:

#### **HeaderA: HeaderB: FooterA: FooterB: PageNumber:**

Each of these parameters specifies whether the corresponding header, footer, or page number will print on the current page. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

#### **PageToBottom:**

This parameter specifies whether the page number will be printed at the bottom center of the current page. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

PageSuppressDlg

---

## PageSuppressDlg

The PageSuppressDlg command displays the Suppress dialog box, which lets the user insert a code that suppresses specified headers, footers, and page numbering codes. This means that the specified headers, footers, and page numbers will not appear on the current page.

The PageSuppressDlg command is the equivalent of choosing Page from the Layout menu, then choosing Suppress.

### **syntax**

PageSuppressDlg ( )

PageSuppress

---

## PageWidowOrphan

Use the PageWidowOrphan command to turn the Widow/Orphan feature on or off. The Widow/Orphan feature prevents one line of a paragraph from being printed by itself at the top or bottom of a page. This command is similar to choosing Page from the Layout menu, then choosing Widow/Orphan.

### **syntax**

PageWidowOrphan (Protect:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **Protect:**

This parameter specifies whether the Widow/Orphan feature should be turned on or off. The valid members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
On!	1

---

## PaperSizeAdd

The PaperSizeAdd command creates a Paper Size/Type definition which can be selected in the Paper Size dialog box. Use Paper Size/Type definitions to specify the type and orientation of paper in the printer. The PaperSizeAdd command is the equivalent of setting options in the Add Paper Size and Edit Labels dialog boxes (these are accessed through the Add... and Edit... buttons on the Paper Size selection in the submenu Page under the Layout menu).

### **syntax**

PaperSizeAdd (Type:Value Set Member; TypeName:Character Expression; FormLength:Measurement Expression; Width:Measurement Expression; TopAdj:Measurement Expression; SideAdj:Measurement Expression; FontOrient:Value Set Member; Loaded:Value Set Member; Location:Numeric Expression; BindEdge:Value Set Member; Duplex:Value Set Member; LabRows:Numeric Expression; LabCols:Numeric Expression; LabLeft:Measurement Expression; LabTop:Measurement Expression; LabWidth:Measurement Expression; LabLength:Measurement Expression; LabBetwnRows:Measurement Expression; LabBetwnCols:Measurement Expression; LabMarLeft:Measurement Expression; LabMarRight:Measurement Expression; LabMarTop:Measurement Expression; LabMarBottom:Measurement Expression)

### **parameters**

This command has 23 parameters:

#### **Type:**

This parameter specifies the type of paper upon which the paper size is based. The members and corresponding numeric equivalents of this value set correspond to the Paper Type pop-up list in the Add Paper Size dialog box:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

AllOthers!	0
Standard!	1
Bond!	2
Letterhead!	3
Labels!	4
Envelope!	5
Transparency!	6
Cardstock!	7
Other!	8

#### **TypeName:**

Use this parameter to specify the paper type name if the Type parameter is set to Other. This character expression is the equivalent of typing a name in the Paper Type group box in the Add Paper Size dialog box.

#### **FormLength:**

This measurement expression sets the length of the page, which corresponds to the length specified in the Paper Size group box in the Add Paper Size dialog box.

#### **Width:**

This measurement expression sets the width of the page, which corresponds to the width specified in the Paper Size group box in the Add Paper Size dialog box.

**TopAdj:**

This measurement expression moves the contents of the page up or down. If the contents of the page should be moved up, this parameter should have a positive value. If the contents of the page should be moved down, this parameter should have a negative value. This parameter corresponds to setting the Top option in the Text Adjustments group box in the Add Paper Size dialog box.

**SideAdj:**

This measurement expression moves the contents of the page to the left or right. If the contents of the page should be moved to the left, this parameter should have a positive value. If the contents of the page should be moved to the right, this parameter should have a negative value. This parameter corresponds to setting the Side option in the Text Adjustments group box in the Add Paper Size dialog box.

**FontOrient:**

This parameter specifies whether fonts should be printed portrait, landscape, or both. This parameter should be set to the Both value only if the Type parameter is set to AllOthers. The valid values are:

Member	Numeric Equivalent
--------	--------------------

Port!	1
Land!	2
Both!	3

**Loaded:**

This parameter specifies whether WordPerfect should prompt the user to insert the correct paper type when printing. This parameter corresponds to either selecting or clearing the Prompt to Load Paper check box in the Add Paper Size dialog box.

Member	Numeric Equivalent
--------	--------------------

No!	0
Yes!	1

**Location:**

This numeric expression specifies whether the form being defined should be printed as continuous, from a particular bin, or by manual feed. If the value of this parameter is 0, the form is set as continuous. If the value of this parameter ranges from 1 to 31, the form is printed from the corresponding bin. If the value of this parameter is 32, the form is printed by manual feed. Any other value sets the form to continuous. This parameter corresponds to the Location pop-up list in the Add Paper Size dialog box.

*Note:* If you are using a Windows printer driver and the value of this parameter is 7, the form is set as auto select (auto select allows the form to feed from the printer's default paper location). If the value of this parameter is 4, the form is set as manual feed. If the value of this parameter is 8, the form will be tractor fed.

**BindEdge:**

This parameter specifies whether a binding width will be applied to the top (short) or left side (long) of the printed page. This parameter corresponds to the Binding pop-up list in the Add Paper Size dialog box. The valid members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Long!	0
Short!	1

**Duplex:**

This command specifies whether the form being defined should be printed on both sides of the page. (This parameter should be set to Yes only if the printer being used has duplex printing capability.) This parameter is the equivalent of either selecting (Yes) or clearing (No) the Double Sided Printing check box in the Add Paper Size dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

No!	0
Yes!	1

**LabRows:**

This numeric expression specifies the number of rows of labels for the paper type. This parameter is the equivalent of setting the Rows value in the Labels Per Page group in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, set this parameter to -1 and omit all subsequent parameters for this command.

**LabCols:**

This numeric expression specifies the number of columns of labels to be used with the paper type being defined. This parameter is the equivalent of setting the Columns value of the Labels Per Page group in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabLeft:**

This measurement expression specifies the distance from the left edge of the page to the left edge of the first column of labels. This parameter is the equivalent of typing a measurement in the Left Edge box in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabTop:**

This measurement expression specifies the distance from the top edge of the page to the top edge of the first row of labels. This parameter is the equivalent of typing a measurement in the Top Edge box in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabWidth:**

This measurement expression specifies the width of an individual label. This parameter is the equivalent of typing a measurement in the Width box in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabLength:**

This measurement expression specifies the height of an individual label. This parameter is the equivalent of typing a measurement in the Height box in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabBetwnRows:**

This measurement expression specifies the amount of blank space between rows of labels. This parameter is the equivalent of typing a measurement in the Rows box of the Distance Between Labels group in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabBetwnCols:**

This measurement expression specifies the amount of blank space between columns of labels. This parameter is the equivalent of typing a measurement in the Columns box of the Distance Between Labels group in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabMarLeft:**

This measurement expression specifies the left margin space for individual labels. This parameter is the equivalent of typing a measurement in the Left box in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabMarRight:**

This measurement expression specifies the right margin space for individual labels. This parameter is the equivalent of typing a measurement in the Right box in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabMarTop:**

This measurement expression specifies the top margin space for individual labels. This parameter is the equivalent of typing a measurement in the Top box in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

**LabMarBottom:**

This measurement expression specifies the Bottom margin space for individual labels. This parameter is the equivalent of typing a measurement in the Bottom box in the Edit Labels dialog box. If the paper type you are defining does not use the Labels feature, omit this parameter.

PaperSizeDelete

PaperSizeDlg

---

# PaperSizeDelete

The PaperSizeDelete command removes a specified Paper Size/Type definition. Since it is possible to have more than one paper type with the same name, this command's parameters are necessary to let you precisely specify which paper type to delete. The PaperSizeDelete command is the equivalent of choosing and deleting a paper type in the Paper Size dialog box.

## **syntax**

PaperSizeDelete (Type:*Value Set Member*; TypeName:*Character Expression*;  
FormLength:*Measurement Expression*; Width:*Measurement Expression*; TopAdj:*Measurement Expression*; SideAdj:*Measurement Expression*; FontOrient:*Value Set Member*; Loaded:*Value Set Member*; Location:*Numeric Expression*; BindEdge:*Value Set Member*; Duplex:*Value Set Member*; LabRows:*Numeric Expression*; LabCols:*Numeric Expression*; LabLeft:*Measurement Expression*; LabTop:*Measurement Expression*; LabWidth:*Measurement Expression*; LabLength:*Measurement Expression*; LabBetwnRows:*Measurement Expression*; LabBetwnCols:*Measurement Expression*; LabMarLeft:*Measurement Expression*; LabMarRight:*Measurement Expression*; LabMarTop:*Measurement Expression*; LabMarBottom:*Measurement Expression*)

## **parameters**

This command has 23 parameters:

### **Type:**

This parameter specifies the type of paper upon which the paper size is based. The members and corresponding numeric equivalents of this value set correspond to the Paper Type pop-up list in the Add Paper Size dialog box:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

AllOthers!	0
Standard!	1
Bond!	2
Letterhead!	3
Labels!	4
Envelope!	5
Transparency!	6
Cardstock!	7
Other!	8

### **TypeName:**

Use this parameter to specify the paper type name if the Type parameter is set to Other. This character expression corresponds to the name in the Paper Type group box in the Add Paper Size dialog box.

### **FormLength:**

This measurement expression specifies the length of the page, which corresponds to the length specified in the Paper Size group box in the Add Paper Size dialog box.

### **Width:**

This measurement expression specifies the width of the page, which corresponds to the



width specified in the Paper Size group box in the Add Paper Size dialog box.

**TopAdj:**

This measurement expression specifies whether the contents of the page are adjusted up or down. This parameter corresponds to the Top option in the Text Adjustments group box in the Add Paper Size dialog box.

**SideAdj:**

This measurement expression specifies whether the contents of the page are adjusted to the left or right. This parameter corresponds to the Side option in the Text Adjustments group box in the Add Paper Size dialog box.

**FontOrient:**

This parameter specifies whether fonts are printed portrait, landscape, or both. This parameter should be set to the Both value only if the Type parameter is set to AllOthers. The valid values are:

Member	Numeric Equivalent
Port!	1
Land!	2
Both!	3

**Loaded:**

This parameter specifies whether WordPerfect prompts the user to insert the correct paper type when printing. This parameter corresponds to selecting or clearing the Prompt to Load Paper check box in the Add Paper Size dialog box.

Member	Numeric Equivalent	Definition
No!	0	The user is prompted to load the paper.
Yes!	1	The user is not prompted to load the paper.

**Location:**

This numeric expression specifies whether the form is printed as continuous, from a particular bin, or by manual feed. If the value of this parameter is 0, the form is set as continuous. If the value of this parameter is from 1 to 31, the form is printed from the corresponding bin. If the value of this parameter is 32, the form is printed by manual feed. This parameter corresponds to the Location pop-up list in the Add Paper Size dialog box.

*Note:* If you are using a Windows printer driver and the value of this parameter is 7, the form is set as auto select (auto select allows the form to feed from the printer's default paper location). If the value of this parameter is 4, the form is set as manual feed. If the value of this parameter is 8, the form will be tractor fed.

**BindEdge:**

This parameter specifies whether binding width is applied to the top (short) or left side (long) of the printed page. This parameter corresponds to the Binding pop-up list in the Add Paper Size dialog box. The valid members of this value set are:

Member	Numeric Equivalent
Long!	0
Short!	1

**Duplex:**

This command specifies whether the form is printed on both sides of the page. This parameter corresponds to the Double Sided Printing check box in the Add Paper Size dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

**LabRows:**

This numeric expression specifies the number of rows of labels for the paper type being deleted. This parameter corresponds to the Rows value in the Labels Per Page group in the Edit Labels dialog box. If the paper type you are deleting does not use the Labels feature, set this parameter to -1 and leave out all subsequent parameters for this command.

**LabCols:**

This numeric expression specifies the number of columns of labels for the paper type being deleted. This parameter corresponds to the Columns value of the Labels Per Page group in the Edit Labels dialog box.

**LabLeft:**

This measurement expression specifies the distance from the left edge of the page to the left edge of the first column of labels. This parameter corresponds to the measurement in the Left Edge box in the Edit Labels dialog box.

**LabTop:**

This measurement expression specifies the distance from the top edge of the page to the top edge of the first row of labels. This parameter corresponds to the measurement in the Top Edge box in the Edit Labels dialog box.

**LabWidth:**

This measurement expression specifies the width of an individual label. This parameter corresponds to a measurement in the Width box in the Edit Labels dialog box.

**LabLength:**

This measurement expression specifies the height of an individual label. This parameter corresponds to the measurement in the Height box in the Edit Labels dialog box.

**LabBetwnRows:**

This measurement expression specifies the amount of blank space between rows of labels. This parameter corresponds to the measurement in the Rows box of the Distance Between Labels group in the Edit Labels dialog box.

**LabBetwnCols:**

This measurement expression specifies the amount of blank space between columns of labels. This parameter corresponds to the measurement in the Columns box of the Distance Between Labels group in the Edit Labels dialog box.

**LabMarLeft:**

This measurement expression specifies the left margin space for individual labels. This parameter corresponds to the measurement in the Left box in the Edit Labels dialog box.

**LabMarRight:**

This measurement expression specifies the right margin space for individual labels. This parameter corresponds to the measurement in the Right box in the Edit Labels dialog box.

**LabMarTop:**

This measurement expression specifies the top margin space for individual labels. This parameter corresponds to the measurement in the Top box in the Edit Labels dialog box.

**LabMarBottom:**

This measurement expression specifies the Bottom margin space for individual labels. This parameter corresponds to the measurement in the Bottom box in the Edit Labels dialog box.

PaperSizeAdd

PaperSizeDlg

PaperSizeSelect

---

## PaperSizeDlg

The PaperSizeDlg command displays the Paper Size dialog box, which lets the user insert a paper size code. If Auto Code Placement is on, the code is inserted at the top of the page in the active document. If Auto Code Placement is off, the form size code is inserted at the insertion point in the active document. The paper size code specifies the size of page to be used from that point forward in the document. This command is the equivalent of choosing Page from the Layout menu, then choosing Paper Size.

### **syntax**

PaperSizeDlg ( )

PaperSizeAdd

PaperSizeDelete

PaperSizeSelect

---

## PaperSizeListCreate

The PaperSizeListCreate command creates a list of paper sizes that are available with the printer currently selected.

### **syntax**

PaperSizeListCreate (Count)

### **parameters**

This command has one parameter.

#### **Count**

This parameter is a variable. The Count parameter is a variable containing the total number of the paper sizes in the list you have created. This parameter is optional.

[PaperSizeSelect](#)

[PaperSizeListFree](#)

[PaperSizeListGetItem](#)

---

## **PaperSizeListFree**

The PaperSizeListFree command frees the list of paper sizes created with the PaperSizeListCreate command.

### **syntax**

PaperSizeListFree ( )

[PaperSizeListCreate](#)

---

## PaperSizeListItem

The PaperSizeListItem command retrieves an item from the list of paper sizes that was created with the PaperSizeListCreate command.

### syntax

PaperSizeListItem (Item: *Numeric Expression*; Type; TypeName; FormLength; Width; TopAdj; SideAdj; FontOrient; Loaded; Location; BindEdge; Duplex; LabRows; LabCols; LabLeft; LabTop; LabWidth; LabLength; LabBetwnRows; LabBetwnCols; LabMarLeft; LabMarRight; LabMarTop; LabMarBottom)

### parameters

This command has several parameters.

#### Item

This parameter is a numeric expression. When the PaperSizeListCreate command creates a list of the paper sizes available, the list is numbered. The Item specifies the number of the paper size to be retrieved from the list that was created.

The following parameters are all variables. Each contains a return value corresponding to the following descriptions:

#### Type:

The Type parameter contains a value for the type of paper upon which the paper size is based. The values from this set correspond to the Paper Type pop-up list in the Add Paper Size dialog box:

Member	Numeric Equivalent
--------	--------------------

AllOthers	0
Standard	1
Bond	2
Letterhead	3
Labels	4
Envelope	5
Transparency	6
Cardstock	7
Other	8

#### TypeName:

The TypeName parameter returns a value for the paper type name if the Type parameter returns a value for Other. This parameter is optional unless the Type parameter returns a value for Other.

#### FormLength:

The FormLength parameter returns a value for the length of the page, in WPunits. This parameter is optional.

#### Width:

The Width parameter returns a value for the width of the page, in WPunits. This parameter is optional.

**TopAdj:**

The TopAdj parameter returns a value indicating whether the contents of the page are adjusted up or down, indicated in WPunits. This parameter is optional.

**SideAdj:**

The SideAdj parameter returns a value indicating whether the contents of the page are adjusted to the left or right, in WPunits. This parameter is optional.

**FontOrient:**

The FontOrient parameter returns a value indicating whether fonts are printed portrait or landscape. This parameter is optional. The values are:

Member	Numeric Equivalent
--------	--------------------

Portrait	1
Landscape	2

**Loaded:**

The Loaded parameter returns a value indicating whether WordPerfect prompts the user to insert the correct paper type when printing. This parameter is optional. The values are:

Member	Numeric Equivalent	Definition
--------	--------------------	------------

No	0	The user is prompted to load the paper.
Yes	1	The user is not prompted to load the paper.

**Location:**

The Location parameter returns a value indicating whether the form is printed as continuous or by manual feed. If the return value of this parameter is 0, the form is set as continuous. If the return value of this parameter is 32, the form is printed by manual feed. This parameter is optional.

*Note:* If you are using a Windows printer driver and the value of this parameter is 7, the form is set as auto select (auto select allows the form to feed from the printer's default paper location). If the value of this parameter is 4, the form is set as manual feed. If the value of this parameter is 8, the form will be tractor fed.

**BindEdge:**

The BindEdge parameter returns a value indicating whether binding width is applied to the top (short) or left side (long) of the printed page. This parameter is optional. The return values are:

Member	Numeric Equivalent
--------	--------------------

Long	0
Short	1

**Duplex:**

The Duplex parameter returns a value indicating whether the form is printed on both sides of the page. This parameter is optional. The return values are:

Member	Numeric Equivalent
--------	--------------------

No	0
Yes	1



**LabRows:**

The LabRows parameter returns a value indicating the number of rows of labels for the paper type being listed. This parameter is optional.

**LabCols:**

The LabCols parameter returns a value indicating the number of columns of labels for the paper type being listed. This parameter is optional.

**LabLeft:**

The LabLeft parameter returns a value indicating the distance from the left edge of the page to the left edge of the first column of labels. This parameter is optional.

**LabTop:**

The LabTop parameter returns a value indicating the distance from the top edge of the page to the top edge of the first row of labels. This parameter is optional.

**LabWidth:**

The LabWidth parameter returns a value indicating the width of an individual label. This parameter is optional.

**LabLength:**

The LabLength parameter returns a value indicating the height of an individual label. This parameter is optional.

**LabBetwnRows:**

The LabBetwnRows parameter returns a value indicating the amount of blank space between rows of labels. This parameter is optional.

**LabBetwnCols:**

The LabBetwnCols parameter returns a value indicating the amount of blank space between columns of labels. This parameter is optional.

**LabMarLeft:**

The LabMarLeft parameter returns a value indicating the left margin space for individual labels. This parameter is optional.

**LabMarRight:**

The LabMarRight parameter returns a value indicating the right margin space for individual labels. This parameter is optional.

**LabMarTop:**

The LabMarTop parameter returns a value indicating the top margin space for individual labels. This parameter is optional.

**LabMarBottom:**

The LabMarBottom parameter returns a value indicating the bottom margin space for individual labels. This parameter is optional.

[PaperSizeListCreate](#)

---

## PaperSizeSelect

The PaperSizeSelect command inserts a code into the active document, specifying that a certain paper type will be used from that point forward in the document. Since it is possible to have more than one paper type with the same name, this command's parameters are necessary to let you specify exactly which paper type to select. The PaperSizeSelect command is the equivalent of selecting a paper type in the Paper Size dialog box.

### syntax

PaperSizeSelect (Type:Value Set Member; TypeName:Character Expression; FormLength:Measurement Expression; Width:Measurement Expression; TopAdj:Measurement Expression; SideAdj:Measurement Expression; FontOrient:Value Set Member; Loaded:Value Set Member; Location:Numeric Expression; BindEdge:Value Set Member; Duplex:Value Set Member; LabRows:Numeric Expression; LabCols:Numeric Expression; LabLeft:Measurement Expression; LabTop:Measurement Expression; LabWidth:Measurement Expression; LabLength:Measurement Expression; LabBetwnRows:Measurement Expression; LabBetwnCols:Measurement Expression; LabMarLeft:Measurement Expression; LabMarRight:Measurement Expression; LabMarTop:Measurement Expression; LabMarBottom:Measurement Expression)

### parameters

This command has 23 parameters:

#### Type:

This parameter specifies the type of paper upon which the paper size is based. The members and corresponding numeric equivalents of this value set correspond to the Paper Type pop-up list in the Add Paper Size dialog box:

Member	Numeric Equivalent
--------	--------------------

AllOthers!	0
Standard!	1
Bond!	2
Letterhead!	3
Labels!	4
Envelope!	5
Transparency!	6
Cardstock!	7
Other!	8

#### TypeName:

Use this parameter to specify the paper type name if the Type parameter is set to Other. This character expression corresponds to the name in the Paper Type group box in the Add Paper Size dialog box.

#### FormLength:

This measurement expression specifies the length of the page, which corresponds to the length specified in the Paper Size group box in the Add Paper Size dialog box.

#### Width:

This measurement expression specifies the width of the page, which corresponds to the

width specified in the Paper Size group box in the Add Paper Size dialog box.

**TopAdj:**

This measurement expression specifies whether the contents of the page are adjusted up or down. This parameter corresponds to the Top option in the Text Adjustments group box in the Add Paper Size dialog box.

**SideAdj:**

This measurement expression specifies whether the contents of the page are adjusted to the left or right. This parameter corresponds to the Side option in the Text Adjustments group box in the Add Paper Size dialog box.

**FontOrient:**

This parameter specifies whether fonts are printed portrait, landscape, or both. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Port!	1
Land!	2
Both!	3

**Loaded:**

This parameter specifies whether WordPerfect prompts the user to insert the correct paper type when printing. This parameter corresponds to selecting or clearing the Prompt to Load Paper check box in the Add Paper Size dialog box.

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

No!	0
Yes!	1

**Location:**

This numeric expression specifies whether the form being defined is printed as continuous, from a particular bin, or by manual feed. If the value of this parameter is 0, the form is set as continuous. If the value of this parameter is from 1 to 31, the form is printed from the corresponding bin. If the value of this parameter is 32, the form is printed by manual feed. This parameter corresponds to the Location pop-up list in the Add Paper Size dialog box.

*Note:* If you are using a Windows printer driver and the value of this parameter is 7, the form is set as auto select (auto select allows the form to feed from the printer's default paper location). If the value of this parameter is 4, the form is set as manual feed. If the value of this parameter is 8, the form will be tractor fed.

**BindEdge:**

This parameter specifies whether binding width will be applied to the top (short) or left side (long) of the printed page. This parameter corresponds to the Binding pop-up list in the Add Paper Size dialog box. The valid members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Long!	0
Short!	1

**Duplex:**

This command specifies whether the form being defined is printed on both sides of the page. (This parameter should only be set to Yes if the printer being used has duplex printing capability.) This parameter corresponds to the Double Sided Printing check box in the Add Paper Size dialog box. The valid values are:

Member	Numeric Equivalent
--------	--------------------

No!	0
-----	---

Yes!	1
------	---

**LabRows:**

This numeric expression specifies the number of rows of labels used for the paper type being selected. This parameter corresponds to setting the Rows value in the Labels Per Page group in the Edit Labels dialog box. If the paper type you are selecting does not use the Labels feature, set this parameter to -1 and leave out all subsequent parameters for this command.

**LabCols:**

This numeric expression specifies the number of columns of labels for the paper type being selected. This parameter corresponds to entering the Columns value of the Labels Per Page group in the Edit Labels dialog box.

**LabLeft:**

This measurement expression specifies the distance from the left edge of the page to the left edge of the first column of labels. This parameter corresponds to entering a measurement in the Left Edge box in the Edit Labels dialog box.

**LabTop:**

This measurement expression specifies the distance from the top edge of the page to the top edge of the first row of labels. This parameter corresponds to the measurement in the Top Edge box in the Edit Labels dialog box.

**LabWidth:**

This measurement expression specifies the width of an individual label. This parameter corresponds to the measurement in the Width box in the Edit Labels dialog box.

**LabLength:**

This measurement expression specifies the height of an individual label. This parameter corresponds to the measurement in the Height box in the Edit Labels dialog box.

**LabBetwnRows:**

This measurement expression specifies the amount of blank space between rows of labels. This parameter corresponds to the measurement in the Rows box of the Distance Between Labels group in the Edit Labels dialog box.

**LabBetwnCols:**

This measurement expression specifies the amount of blank space between columns of labels. This parameter corresponds to the measurement in the Columns box of the Distance Between Labels group in the Edit Labels dialog box.

**LabMarLeft:**

This measurement expression specifies the left margin space for individual labels. This parameter corresponds to the measurement in the Left box in the Edit Labels dialog box.

**LabMarRight:**

This measurement expression specifies the right margin space for individual labels. This parameter corresponds to the measurement in the Right box in the Edit Labels dialog box.

**LabMarTop:**

This measurement expression specifies the top margin space for individual labels. This parameter corresponds to the measurement in the Top box in the Edit Labels dialog box.

**LabMarBottom:**

This measurement expression specifies the Bottom margin space for individual labels. This parameter corresponds to the measurement in the Bottom box in the Edit Labels dialog box.

PaperSizeAdd

PaperSizeDelete

PaperSizeDlg

---

## ParagraphDoubleIndent

The ParagraphDoubleIndent command indents both sides of the paragraph. This command is the equivalent of choosing Double Indent under Paragraph on the Layout menu.

### **syntax**

ParagraphDoubleIndent ( )

---

## ParagraphHangingIndent

The ParagraphHangingIndent command leaves the first line of a paragraph flush with the left margin and indents all remaining lines to the next tab stop. This command is the equivalent of choosing Hanging Indent under Paragraph on the Layout menu.

### **syntax**

ParagraphHangingIndent ( )

---

## ParagraphIndent

The ParagraphIndent command indents all lines of a paragraph to the next tab stop. (A tab pushes only one line to the next tab stop). This command is the equivalent of choosing Indent under Paragraph on the Layout menu.

### **syntax**

ParagraphIndent ( )



---

## ParagraphNumber

The ParagraphNumber command inserts a paragraph numbering code at the insertion point in the active document, either for the paragraph numbering level automatically determined by WordPerfect, or for a specified paragraph numbering level. This command is the equivalent of selecting Auto or Manual in the Paragraph Numbering dialog box, then, if Manual is selected, typing a level number.

### **syntax**

ParagraphNumber (Level:*Numeric Expression*; InsertType:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **Level:**

This numeric expression specifies the level of paragraph number to insert into the document. You may omit this parameter if the InsertType parameter is set to Auto!.

#### **InsertType:**

This parameter specifies whether WordPerfect should automatically determine the paragraph numbering level to insert into the document, or whether the paragraph numbering level is specified in the Level parameter. If no parameters are specified for the ParagraphNumber command, the paragraph numbering level is determined automatically. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Auto!	0
-------	---

Fixed!	1
--------	---

ParagraphNumberingDlg

---

## ParagraphNumberDefDlg

The ParagraphNumberDefDlg command displays the Define Paragraph Numbering dialog box, which lets the user define paragraph levels, the outline starting number, the outline style, and outline options. This command is the equivalent of choosing Outline from the Tools menu, then choosing Define.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

ParagraphNumberDefDlg ( )

---

## ParagraphNumberingDlg

The ParagraphNumberingDlg displays the Paragravph Numbering dialog box, which lets the user choose whether WordPerfect automatically determines paragraph numbering or whether the user specifies paragraph numbering. This command is the equivalent of choosing Outline from the Tools menu, then choosing Paragraph Number.

### **syntax**

ParagraphNumberingDlg ( )

---

## PauseKey

The PauseKey command temporarily stops the macro, allowing the user to use WordPerfect as if there were no macro running. When the user presses a specified key, the macro resumes playing at the command immediately following the PauseKey command.

Note: This command is not the same as the PAUSE programming command. The PAUSE programming command requires that the user choose Pause from the Macro menu to continue macro play.

### **syntax**

PauseKey (Key:*Value Set Member*; Character:*Character Expression*)

### **parameters**

This command has two parameters:

#### **Key:**

This parameter specifies whether pressing Enter (Return), Cancel (Esc), Close (Ctrl+F4), a specified character , or any key will cause the macro to resume playing. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Enter!	0
Cancel!	1
Close!	2
Character!	3
Any!	4

If the Character value (3 above) is specified, the character specified in the Character parameter, below, will cause the macro to resume playing.

#### **Character:**

This character expression specifies the character that, when pressed, will cause the macro to resume playing. This parameter is case sensitive. The character expression should contain one character. If it contains more than one, only the first is recognized. If the Key parameter is not set to Character!, this parameter is ignored and can be omitted.

---

## PosCellDown

The PosCellDown command moves the insertion point down to the next row in the table. This command will cause an error message box to appear if you are not in a table. The PosCellDown command is the equivalent of pressing Alt+Down Arrow.

### **syntax**

PosCellDown ( )

PosCellUp

SelectCellDown

---

## PosCellNext

The PosCellNext command moves the insertion point right to the next column in the table. If the insertion point is at the rightmost column in the table, the insertion point is moved to the leftmost column in the table, one row down. If the insertion point is not in a table, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. This command will not move the insertion point outside of a table. The PosCellNext command is the equivalent of pressing Alt+Right Arrow or Tab.

This command is not recordable. To use the PosCellNext command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Alt+Right Arrow or Tab, the macro facility records the PosColumnNext ( ) or Tab ( ) command, respectively, which will perform the same function.

### **syntax**

PosCellNext ( )

LABEL

ONERROR

PosCellPrevious

SelectCellRight

---

## PosCellPrevious

The PosCellPrevious command moves the insertion point left to the previous column in the table. If the insertion point is at the leftmost column in the table, the insertion point is moved to the rightmost column in the table, one row up. If the insertion point is not in a table, this command terminates macro execution or redirects it to the LABEL specified by ONERROR. This command will not move the insertion point outside of a table.

This command is not recordable. To use the PosCellPrevious command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Alt+Left Arrow or Shift+Tab, the macro facility records the PosColumnPrevious ( ) or MarginRelease ( ) command, respectively, which will perform the same function.

### **syntax**

PosCellPrevious ( )

LABEL

ONERROR

PosCellNext

SelectCellLeft

---

## PosCellUp

The PosCellUp command moves the insertion point up to the next row in the table. This command will cause an error message box to appear if you are not in a table. The PosCellUp command is the equivalent of pressing Alt+Up Arrow.

### **syntax**

PosCellUp ( )

PosCellDown

SelectCellUp



---

## **PosCharNext**

The PosCharNext command moves the insertion point one character to the right. If the insertion point is at the end of the document, this command has no effect. This command is the equivalent of pressing Right Arrow.

### **syntax**

PosCharNext ( )

PosCharPrevious

---

## PosCharPrevious

The PosCharPrevious command moves the insertion point one character to the left. If the insertion point is at the very beginning of the document, this command has no effect. This command is the equivalent of pressing Left Arrow.

### **syntax**

PosCharPrevious ( )

PosCharNext

---

## PosColumnNext

The PosColumnNext command moves the insertion point one column to the right. If columns are not on at the insertion point, this command has no effect. This command is the equivalent of pressing Alt+Right Arrow in columns.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosColumnNext ( )

[PosColumnPrevious](#)

---

## PosColumnPrevious

The PosColumnPrevious command moves the insertion point one column to the left. If columns are not on at the insertion point, this command has no effect. This command is the equivalent of pressing Alt+Left Arrow in columns.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosColumnPrevious ( )

PosColumnNext

---

## **PosDocBottom**

The PosDocBottom command moves the insertion point to the end of the active document. This command is the equivalent of pressing Ctrl+End.

### **syntax**

PosDocBottom ( )

PosDocTop

---

## **PosDocTop**

The PosDocTop command moves the insertion point to the beginning of text in the active document. This command is the equivalent of pressing Ctrl+Home.

### **syntax**

PosDocTop ( )

PosDocBottom

PosDocVeryTop

---

## PosDocVeryTop

The PosDocVeryTop command moves the insertion point to the very beginning of the active document. The insertion point will be placed before any text or codes in the document.

This command is not recordable. To use the PosDocVeryTop command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Ctrl+Home, Ctrl+Home, the macro facility records the PosDocTop ( ) command twice.

### **syntax**

PosDocVeryTop ( )

PosDocBottom

PosDocTop

SelectDocVeryTop

---

## PosGoTo

The PosGoTo command moves the insertion point to a specified page, to the top or bottom of a page, or to a specific location within a column or table. This command is the equivalent of setting options in the Go To dialog box.

### **syntax**

PosGoTo (Where:*Value Set Member*; Page:*Numeric Expression*; Cell:*Character Expression*)

### **parameters**

This command has three parameters:

#### **Where:**

This parameter specifies the type of movement the PosGoTo command will perform. This parameter should be left blank if the Page or Cell parameter is used.

Not all of the options listed below are valid at any given time. For instance, the BeginningOfSelection! value is valid only when something is currently selected. The options below correspond to the options available in the Position pop-up list in the Go To dialog box.

The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

BeginningOfSelection!	0
ReselectText!	1
TopOfPage!	2
BottomOfPage!	3
TopOfColumn!	4
BottomOfColumn!	5
PreviousColumn!	6
NextColumn!	7
FirstColumn!	8
LastColumn!	9
TopOfCell!	10
BottomOfCell!	11
FirstCell!	12
LastCell!	13
LastPosition!	14

#### **Page:**

This numeric expression specifies a page number to which the insertion point is sent. This parameter should be left blank if the Where or Cell parameter is used.

#### **Cell:**

This character expression specifies the row and column in a table to which the insertion point is sent. (For example, PosGoTo (Cell: "B4") would send the insertion point to cell B4.) This parameter should be left blank if the Where or Page parameter is used.



PosGoToDlg

---

## PosGoToDlg

The PosGoToDlg command displays the Go To dialog box, which lets the user specify where to place the insertion point. The PosGoToDlg command is the equivalent of choosing Go To from the Edit menu.

### **syntax**

PosGoToDlg ( )

PosGoTo

---

## PosLineBegin

The PosLineBegin command moves the insertion point to the beginning of the text on the current line. This command is the equivalent of pressing Home.

### **syntax**

PosLineBegin ( )

PosLineEnd

PosLineVeryBegin

---

## **PosLineDown**

The PosLineDown command moves the insertion point one line down. If the insertion point is already at the last line in the document, this command has no effect. The PosLineDown command is the equivalent of pressing Down Arrow in a document window.

### **syntax**

PosLineDown ( )

PosLineUp

---

## PosLineEnd

The PosLineEnd command moves the insertion point to the end of the current line. This command is the equivalent of pressing End.

### **syntax**

PosLineEnd ( )

PosLineBegin

PosLineVeryBegin

---

## PosLineUp

The PosLineUp command moves the insertion point one line up. If the insertion point is already at the first line in the document, this command has no effect. The PosLineUp command is the equivalent of pressing Up Arrow in a document window.

### **syntax**

PosLineUp ( )

PosLineDown

---

## PosLineVeryBegin

The PosLineVeryBegin command moves the insertion point to the beginning of the current line, in front of all codes and text.

This command is not recordable. To use the PosLineVeryBegin command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Home,Home, the macro facility records the PosLineBegin ( ) command twice.

### **syntax**

PosLineVeryBegin ( )

PosLineBegin

PosLineEnd

PosLineVeryEnd

---

## PosLineVeryEnd

The PosLineVeryEnd command moves the insertion point to the end of the current line, after all codes and text.

This command is not recordable. To use the PosLineVeryEnd command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing End,End, the macro facility records the PosLineEnd ( ) command twice.

### **syntax**

PosLineVeryEnd ( )

PosLineBegin

PosLineVeryBegin

SelectLineVeryEnd



---

## PosPageBottom

The PosPageBottom command moves the insertion point to the bottom of the current column. If the insertion point is in a table, this command moves the insertion point to the bottom of the cell. Otherwise, this command moves the insertion point to the bottom of the document. This command is the equivalent of pressing Alt+End.

### **syntax**

PosPageBottom ( )

PosPageTop

---

## **PosPageNext**

The PosPageNext command moves the insertion point to the beginning of the following page in the active document. If the insertion point is already in the last page, the PosPageNext command moves the insertion point to the end of the document. This command is the equivalent of pressing Alt+PgDn.

### **syntax**

PosPageNext ( )

PosPagePrevious

---

## PosPagePrevious

The PosPagePrevious command moves the insertion point to the beginning of the previous page in the active document. If the insertion point is already in the first page, this command will move the insertion point to the top of the document. The PosPagePrevious command is the equivalent of pressing Alt+PgUp.

### **syntax**

PosPagePrevious ( )

PosPageNext

---

## **PosPageTop**

The PosPageTop command moves the insertion point to the top of the current column. If the insertion point is in a table, this command moves the insertion point to the top of the cell. Otherwise, this command moves the insertion point to the top of the document. This command is the equivalent of pressing Alt+Home.

### **syntax**

PosPageTop ( )

PosPageBottom

---

## **PosParagraphNext**

The PosParagraphNext command moves the insertion point to the beginning of the next paragraph. If the insertion point is in the last paragraph in the document, this command moves the insertion point to the end of the document. The PosParagraphNext command is the equivalent of pressing Ctrl+Down Arrow.

### **syntax**

PosParagraphNext ( )

[PosParagraphPrevious](#)

---

## **PosParagraphPrevious**

The PosParagraphPrevious command moves the insertion point to the beginning of the current paragraph. If the insertion point is already at the beginning of a paragraph, this command moves the insertion point to the beginning of the preceding paragraph. The PosParagraphPrevious command is the equivalent of pressing Ctrl+Up Arrow.

### **syntax**

PosParagraphPrevious ( )

PosParagraphNext

---

## PosScreenDown

The PosScreenDown command moves the insertion point to the bottom of the screen the first time it is used, then down one screen at a time every time it is used immediately thereafter. Screen, as used in this context means the amount of the window viewable on your terminal screen. The amount of text each "screen down" will reveal depends on how large you have made your window. This command is the equivalent of pressing PgDn (Page Down).

### **syntax**

PosScreenDown ( )

PosScreenUp

---

## PosScreenLeft

The PosScreenLeft command moves the insertion point to the left side of the screen the first time it is used, then left one screen at a time (or to the left margin of the document) every time it is used immediately thereafter. Screen, as used in this context means the amount of the window viewable on your terminal screen. The amount of text each "screen left" will reveal depends on how large you have made your window.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosScreenLeft ( )

PosScreenRight

SelectScreenLeft



---

## PosScreenRight

The PosScreenRight command moves the insertion point to the right side of the screen the first time it is used, then right one screen at a time (or to the right margin of the document) every time it is used immediately thereafter. Screen, as used in this context means the amount of the window viewable on your terminal screen. The amount of text each "screen right" will reveal depends on how large you have made your window.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosScreenRight ( )

PosScreenLeft

SelectScreenRight

---

## PosScreenUp

The PosScreenUp command moves the insertion point to the top of the screen the first time it is used, then up one screen at a time every time it is used immediately thereafter. Screen, as used in this context means the amount of the window viewable on your terminal screen. The amount of text each "screen up" will reveal depends on how large you have made your window. This command is the equivalent of pressing PgUp (Page Up).

### **syntax**

PosScreenUp ( )

PosScreenDown

---

## PosTableBegin

The PosTableBegin command moves the insertion point to the first cell of the current table. This command does not have any effect outside of tables. The PosTableBegin command is the equivalent of choosing First Cell from the Position pop-up list in the Go To dialog box when the insertion point is in a table.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosTableBegin ( )

PosTableColumnBottom

PosTableColumnTop

PosTableEnd

PosTableRowBegin

PosTableRowEnd

---

## PosTableColumnBottom

The PosTableColumnBottom command moves the insertion point to the bottom cell of the current column. This command does not have any effect outside of tables. The PosTableColumnBottom command is the equivalent of choosing Bottom of Column from the Position pop-up list in the Go To dialog box.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosTableColumnBottom ( )

PosGoTo

PosTableColumnTop

PosTableBegin

PosTableEnd

---

## PosTableColumnTop

The PosTableColumnTop command moves the insertion point to the top cell of the current column. This command does not have any effect outside of tables. The PosTableColumnTop command is the equivalent of choosing Top of Column from the Position pop-up list in the Go To dialog box.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosTableColumnTop ( )

PosGoTo

PosTableBegin

PosTableColumnBottom

PosTableEnd

---

## PosTableEnd

The PosTableEnd command moves the insertion point to the last cell (bottom right) of the current table. This command does not have any effect outside of tables. The PosTableEnd command is the equivalent of choosing Last Cell from the Position pop-up list in the Go To dialog box when the insertion point is in a table.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosTableEnd ( )

PosGoTo

PosTableBegin

PosTableColumnBottom

PosTableColumnTop

PosTableRowBegin

PosTableRowEnd

---

## PosTableRowBegin

The PosTableRowBegin command moves the insertion point to the leftmost cell of the current row in the current table. This command does not have any effect outside of tables. The PosTableRowBegin command is the equivalent of pressing Home,Home when the insertion point is in a table.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosTableRowBegin ( )

PosGoTo

PosTableBegin

PosTableColumnBottom

PosTableColumnTop

PosTableEnd

PosTableRowEnd

---

## PosTableRowEnd

The PosTableRowEnd command moves the insertion point to the rightmost cell of the current row in the current table. This command does not have any effect outside of tables. The PosTableRowEnd command is the equivalent of pressing End,End when the insertion point is in a table.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PosTableRowEnd ( )

PosGoTo

PosTableBegin

PosTableColumnBottom

PosTableColumnTop

PosTableEnd

PosTableRowBegin



---

## **PosWordNext**

The PosWordNext command moves the insertion point to the beginning of the next word. If the insertion point is in the final word of the document, this command moves the insertion point to the end of that word. The PosWordNext command is the equivalent of pressing Ctrl+Right Arrow.

### **syntax**

PosWordNext ( )

[PosWordPrevious](#)

---

## **PosWordPrevious**

The PosWordPrevious command moves the insertion point to the beginning of the current word. If the insertion point is at the beginning of the current word, the PosWordPrevious command moves the insertion point to the beginning of the previous word. The PosWordPrevious command is the equivalent of pressing Ctrl+Left Arrow.

### **syntax**

PosWordPrevious ( )

PosWordNext

---

## PrefBackup

The PrefBackup command lets you specify whether WordPerfect makes backup copies of open documents and lets you specify the time interval between backups. This command is the equivalent of setting options in the Backup [dialog box](#).

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefBackup command should be followed by a PrefSave command at some point in the macro . Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefBackup (TimedBackup:*Value Set Member*; MinutesBetween:*Numeric Expression*; OriginalDocBackup:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **TimedBackup:**

This parameter specifies whether copies of the open documents will automatically be saved every x minutes (where x is the number specified in the MinutesBetween parameter). The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

#### **MinutesBetween:**

This numeric expression specifies the number of minutes between automatic backups. This parameter is ignored and can be omitted if the TimedBackup parameter is set to Off!.

#### **OriginalDocBackup:**

This parameter specifies whether the Original Document Backup feature is on. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

[PrefBackupDlg](#)

[PrefSave](#)

---

## **PrefBackupDlg**

The PrefBackupDlg command displays the Backup dialog box, which lets the user choose whether WordPerfect makes backup copies of open documents. Users can also specify how frequently backup copies of open documents . The PrefBackupDlg command is the equivalent of choosing Preferences from the file menu, then choosing Backup.

### **syntax**

PrefBackupDlg ( )

PrefBackup

---

## PrefBeep

The PrefBeep command specifies when WordPerfect will beep. This command is the equivalent of setting options in the Beep On group box in the Environment Settings dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefBeep command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefBeep (*Error:Value Set Member*; *Hyphenation:Value Set Member*; *SearchFailure:Value Set Member*)

### **parameters**

This command has three parameters:

#### **Error:**

This parameter specifies whether WordPerfect will beep when an error occurs. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
On!	1

#### **Hyphenation:**

This parameter specifies whether WordPerfect will beep when the user is required to position a hyphen. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
On!	1

#### **SearchFailure:**

This parameter specifies whether WordPerfect will beep when a search fails (when the item searched for is not found). The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
On!	1

PrefEnvSettingsDlg

PrefSave

---

## PrefDateFormat

The PrefDateFormat command specifies the way the Date feature formats the date in WordPerfect. This command is similar to setting options in the Preferences Date/Time Format [dialog box](#).

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefDateFormat command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefDateFormat (FormatString:Character Expression)

### **parameters**

This command has one parameter:

#### **FormatString:**

This character expression specifies how dates appear when you insert them using the Date feature. However, the method of formatting the date is different than in the Preferences Date/Time Format dialog box.

In the list below, the numbers 0 through 9 represent ways the parts of the date and time can be written. You can include words, spaces, and other punctuation with these numbers to create a date format.

<b>Character</b>	<b>Meaning</b>
1	Day of the Month
2	Month (number)
3	Month (word)
4	Year (all four digits)
5	Year (last two digits)
6	Day of the Week (word)
7	Hour (24-hour clock)
8	Hour (12-hour clock)
9	Minute
0	am / pm
	Used before a number, will pad numbers less than 10 with a leading zero or space, or abbreviate the month or day of the week

Examples:    3 1, 4            =    January 1, 1992  
              %6 %3 1, 4        =    Wed Jan 1, 1992  
              %2/%1/5 (6)    =    01/01/92 (Wednesday)  
              \$2/\$1/5 (%6)    =    1/ 1/92 (Wed)  
              8:90            =    10:55am

DateCode

DateFormat

DateText

PrefDateFormatDlg

PrefSave

---

## **PrefDateFormatDlg**

The PrefDateFormatDlg command displays the Preferences: Date/Time Format dialog box, which lets the user specify how the date and time will appear in documents. The PrefDateFormatDlg command is the equivalent of choosing Preferences from the File menu, then choosing Date Format.

### **syntax**

PrefDateFormatDlg ( )

DateFormatDlg

DateFormat

PrefDateFormat



---

## **PrefDisplayDlg**

The PrefDisplayDlg command displays the Display Settings dialog box, which lets the user specify how text, codes, and graphics will appear in WordPerfect. The PrefDisplayDlg command is the equivalent of choosing Preferences from the File menu, then choosing Display.

### **syntax**

PrefDisplayDlg ( )

PrefDisplaySet

---

## PrefDisplaySet

The PrefDisplaySet command changes the way WordPerfect features are displayed in documents. This command is the equivalent of setting options in the Display Settings dialog box.

If you want these preference options to remain set in the following sessions of WordPerfect, the PrefDisplaySet command should be followed by a PrefSave command at some point in the macro. Otherwise, these options will revert to their former state when you quit WordPerfect.

### syntax

PrefDisplaySet (*TextInSystemColors:Value Set Member; MonochromeGraphics:Value Set Member; AutoFormat:Value Set Member; ColumnsSideBySide:Value Set Member; DispMergeCodes:Value Set Member; SculpturedDialogs:Value Set Member; VerticalScrollBar:Value Set Member; HorizontalScrollBar:Value Set Member; HardReturnChar:Character Expression; DisplayUnits:Value Set Member; StatusBarUnits:Value Set Member; RevealCodesSize: Numeric Expression*)

### parameters

This command has 12 parameters:

**TextInSystemColors: MonochromeGraphics: AutoFormat: ColumnsSideBySide: DispMergeCodes: SculpturedDialogs: VerticalScrollBar: HorizontalScrollBar: RevealCodesSize:**

These parameters set the defaults for the Document Window and Scroll Bar group boxes in the Display Settings dialog box. The valid values are:

Member	Numeric Equivalent	Definition
No!	0	The option is not activated.
Yes!	1	The option is activated.

### HardReturnChar:

This character expression specifies how hard return codes will appear in documents. This character expression should contain only one character. If you do not want a hard return character, use a space for this character expression.

### DisplayUnits:

This parameter sets the default units of measure used for specifying margins, tabs, and so forth. The members of this value set are:

Member	Numeric Equivalent
InchesQuote!	0
Inches!	1
Centimeters!	2
Points!	3
WPU!	4

A point is equal to 1/72", while a WordPerfect unit (WPU) is equal to 1/1200".

### StatusBarUnits:

This parameter sets the default units of measure used on the status bar. The members of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

InchesQuote!	0
Inches!	1
Centimeters!	2
Points!	3
WPU!	4

A point is equal to 1/72", while a WordPerfect unit (WPU) is equal to 1/1200".

### **RevealCodesSize**

This parameter is a numeric expression and sets the size of the Reveal Codes portion of the document window.

PrefDraftColors

PrefRevealCodesColors

PrefSave

PrefZoom

---

## PrefDocSummary

The PrefDocSummary command lets the user specify the subject search text and default description type for WordPerfect document. Users can also specify whether WordPerfect should create a document summary when documents are saved or closed. This command is the equivalent of setting options in the Document Summary Preferences [dialog box](#).

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefDocSummary command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefDocSummary (SubjectSearchText:*Character Expression*; DefaultDocType:*Character Expression*; CreateOnExit:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **SubjectSearchText:**

This character expression specifies the text that precedes the subject line for a document (such as RE:). This parameter is the equivalent of specifying the Subject Search Text value in the Document Summary Preferences dialog box.

#### **DefaultDocType:**

This character expression specifies the text that will automatically appear in the Descriptive Type entry every time you create a document summary. This parameter is the equivalent of typing the Default Descriptive Type in the Document Summary Preferences dialog box.

#### **CreateOnExit:**

This parameter specifies whether WordPerfect will create a document summary when documents are saved or closed. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

[PrefDocSummaryDlg](#)

[PrefSave](#)

---

## **PrefDocSummaryDlg**

The PrefDocSummaryDlg command displays the Document Summary Preferences dialog box, which lets the user specify whether a summary should be created when documents are saved, and specify the subject search text and default document type. The PrefDocSummaryDlg command is the equivalent of choosing Preferences from the File menu, then choosing Document Summary.

### **syntax**

PrefDocSummaryDlg ( )

PrefDocSummary

---

## PrefDraftColors

The PrefDraftColors command sets a foreground and background color for a specified attribute in draft mode. This command is the equivalent of choosing an appearance or size attribute, then selecting a color from the foreground palette and a color from the background palette in the Draft Mode Colors dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefDraftColors command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefDraftColors (FontAttribute:*Value Set Member*; Foreground:*Value Set Member*;  
Background:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **FontAttribute:**

This parameter specifies the appearance or size attribute to which the colors in the Foreground and Background parameters apply. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Normal!	0
Blocked!	1
Underline!	2
Strikeout!	3
Bold!	4
DoubleUnderline!	5
Redline!	6
Shadow!	7
Italics!	8
SmallCaps!	9
Outline!	10
Subscript!	11
Superscript!	12
Fine!	13
Small!	14
Large!	15
VeryLarge!	16
ExtraLarge!	17
BoldUnderline!	18
Combinations!	19

**Foreground:**

This parameter specifies the foreground color for the attribute specified in the FontAttribute parameter. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

White!	0
DkGray!	1
LtRed!	2
LtYellow!	3
LtGreen!	4
LtCyan!	5
LtBlue!	6
LtMagenta!	7
LtGray!	8
Black!	9
DkRed!	10
DkYellow!	11
DkGreen!	12
DkCyan!	13
DkBlue!	14
DkMagenta!	15

**Background:**

This parameter specifies the background color for the attribute specified in the FontAttribute parameter. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

White!	0
DkGray!	1
DkCyan!	2
DkMagenta!	3
LtGray!	4
Black!	5
DkBlue!	6
LtRed!	7

PrefDraftColorsDlg

PrefSave

---

## **PrefDraftColorsDlg**

The PrefDraftColorsDlg command displays the Draft Mode Colors dialog box, which lets the user specify colors to represent attributes when WordPerfect is in draft mode. The PrefDraftColorsDlg command is the equivalent of choosing Preferences from the File menu, choosing Display, then choosing the Draft Mode Colors button in the Display Settings dialog box.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PrefDraftColorsDlg ( )

PrefDraftColors

PrefRevealCodesColorsDlg



---

## PrefEnvSettings

The PrefEnvSettings command lets you specify options that determine how certain WordPerfect features operate. These options include: Auto Code Placement, Confirm on Code Deletion, Fast Save, Allow Undo, Allow Graphic Boxes to Bump to Next Page, and Reformat Documents for Default Printer on Open. This command is the equivalent of setting the Settings options in the Environment Settings [dialog box](#) under Preferences on the File menu.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefEnvSettings command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### syntax

PrefEnvSettings (AutoCodePlacement:*Value Set Member*; ConfirmCodeDeletion:*Value Set Member*; FastSave:*Value Set Member*; AllowUndo:*Value Set Member*; FormatForPrinter:*Value Set Member*; GraphicsBumping:*Value Set Member*)

### parameters

This command has five parameters:

#### AutoCodePlacement:

This parameter specifies whether certain formatting [codes](#), such as margin and justification codes, will automatically be placed at the top of the paragraph or page. The valid values are:

Member	Numeric Equivalent
Off!	0
On!	1

#### ConfirmCodeDeletion:

This parameter specifies whether WordPerfect will display a [prompt](#) for confirmation when Reveal Codes is not on and the user attempts to delete a code. The valid values are:

Member	Numeric Equivalent
No!	0
Yes!	1

#### FastSave:

This parameter specifies whether WordPerfect will save documents without formatting them for printing. If this parameter is set to On!, documents are saved more quickly. The valid values are:

Member	Numeric Equivalent
Off!	0
On!	1

#### AllowUndo:

This parameter specifies whether the Undo feature will be available in WordPerfect. The

valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

**FormatForPrinter:**

This parameter specifies whether documents will automatically be formatted for the default printer when they are opened. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

**GraphicsBumping:**

This parameter specifies whether a page anchor graphics box can be bumped to the next page if a user has inserted additional text or another graphics box above the original graphics box. If this parameter is set to No!, the page anchor graphics box is forced to remain on the current page. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

PrefEnvSettingsDlg

PrefSave

---

## **PrefEnvSettingsDlg**

The PrefEnvSettingsDlg command displays the Environment Settings dialog box, which lets the user specify how beep, hyphenation, the ruler, and other WordPerfect settings will work. The PrefEnvSettingsDlg command is the equivalent of choosing Preferences from the File menu, then choosing Environment.

### **syntax**

PrefEnvSettingsDlg ( )

PrefEnvSettings

---

## PrefEquation

The PrefEquation command specifies the keyboard, font size, alignment, and printing format for equations. This command is the equivalent of setting options in the Equation Preferences Settings [dialog box](#).

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefEquation command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### syntax

PrefEquation (*PrintAsGraphics:Value Set Member; FontSize:Numeric Expression; HorizontalAlignment:Value Set Member; VerticalAlignment:Value Set Member; Keyboard:Character Expression*)

### parameters

This command has four parameters:

#### PrintAsGraphics:

This parameter is the equivalent of turning Print as Graphics on or off. The valid values are:

Member	Numeric Equivalent
--------	--------------------

Text!	0
-------	---

Graphics!	1
-----------	---

#### FontSize:

This numeric expression sets the graphic font size for equations. This parameter is the equivalent of typing a number in the Point Size [text box](#).

#### HorizontalAlignment:

This parameter sets the horizontal alignment for equations. This parameter is the equivalent of choosing an alignment from the Horizontal pop-up list. The members of this value set are:

Member	Numeric Equivalent
--------	--------------------

Left!	0
-------	---

Right!	1
--------	---

Center!	2
---------	---

#### VerticalAlignment:

This parameter sets the vertical alignment for equations. This parameter is the equivalent of choosing an alignment from the Vertical pop-up list. The members of this value set are:

Member	Numeric Equivalent
--------	--------------------

Top!	0
------	---

Center!	1
---------	---

Bottom!	2
---------	---

#### Keyboard:

This character expression specifies the filename of the keyboard to be used with equations. This parameter is the equivalent of selecting a keyboard in the Select Keyboard File dialog box.

EquationEdit

EquationSettings

PrefEquationDlg

PrefSave

---

## **PrefEquationDlg**

The PrefEquationDlg command displays the Equation Preferences dialog box, which lets the user specify the keyboard, font size, and alignment for equations in WordPerfect. The PrefEquationDlg command is the equivalent of choosing Preferences from the File menu, then choosing Equations.

### **syntax**

PrefEquationDlg ( )

EquationEdit

EquationSettings

PrefEquation

---

## PrefHyphenation

The PrefHyphenation command specifies whether hyphenation should be turned on in WordPerfect, and if so, when you should be prompted to insert a hyphen. This command also specifies whether WordPerfect uses an external dictionary or an internal algorithm to determine the correct place to hyphenate. This command is the equivalent of choosing options in the Prompt for Hyphenation and Hyphenation group boxes in the Environment Settings dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefHyphenation command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefHyphenation (Prompt:*Value Set Member*; HyphType:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **Prompt:**

This parameter specifies when and if you should be prompted to place a hyphen. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Never!	0
WhenRequired!	1
Always!	2

#### **HyphType:**

This parameter specifies whether hyphenation should use an external dictionary or an internal algorithm for determining the correct place to hyphenate. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

External!	0
Internal!	1

PrefEnvSettingsDlg

PrefSave

---

## PrefInitialCodes

The PrefInitialCodes command opens the Initial Codes Editor. Once this editor is open, following macro commands apply to the Initial Codes Editor until a Close command is encountered.

The codes placed in the Initial Codes Editor apply to all new documents created in WordPerfect. The PrefInitialCodes command is the equivalent of choosing Preferences from the File menu, then choosing Initial Codes.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefInitialCodes command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefInitialCodes ( )

DocInitialCodes

PrefSave



---

## **PrefLocationDlg**

The PrefLocationDlg command displays the Location of Files dialog box, which lets the user specify directories for storing the various types of files (for instance, document, graphic, and printer files). The PrefLocationDlg command is the equivalent of choosing Preferences from the File menu, then choosing Location of Files.

### **syntax**

PrefLocationDlg ( )

PrefLocationOfFiles

---

## PrefLocationOfFiles

The PrefLocationOfFiles command specifies the default directory location for backup, document, graphic, printer, spreadsheet, macro, style, and dictionary files. This command can also update the Quick List with any changes made. This command is the equivalent of specifying directories in the Location of Files dialog box. However, if the specified directory does not exist, an error message is displayed, rather than a message box prompting the user to create the directory, as in the Location of Files dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefLocationofFiles command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### syntax

PrefLocationOfFiles (BackupFiles:*Character Expression*; Documents:*Character Expression*; GraphicFiles:*Character Expression*; PrinterFiles:*Character Expression*; Spreadsheets:*Character Expression*; MacroFiles:*Character Expression*; StyleFiles:*Character Expression*; StyleLibraryName:*Character Expression*; MainDictFiles:*Character Expression*; SupDictFiles:*Character Expression*; UpdateQuickList:*Value Set Member*)

### parameters

This command has 11 parameters:

#### **BackupFiles: Documents: GraphicFiles: PrinterFiles: Spreadsheets: MacroFiles: StyleFiles:**

These character expressions specify the directory in which their respective file types should be located.

#### **StyleLibraryName:**

This character expression specifies the filename of the default WordPerfect style library.

#### **MainDictFiles: SupDictFiles:**

These character expressions specify the directory in which their respective files should be located.

#### **UpdateQuickList:**

This parameter specifies whether to update Macro, Graphics, and Printer directories in your QuickList with any new directories specified in Location of Files while keeping quick list names intact. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

PrefLocationDlg

PrefSave

---

## PrefMenu

The PrefMenu command specifies whether shortcut keys will appear on the menus and whether the four most recently opened documents will appear at the bottom of the File menu in WordPerfect. This command is the equivalent of setting options in the Menu group box in the Environment Settings dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefMenu command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefMenu (DisplayShortcuts:*Value Set Member*; DisplayLastOpened:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **DisplayShortCuts:**

This parameter specifies whether keystroke shortcuts will be displayed alongside menu items. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

#### **DisplayLastOpened:**

This parameter specifies whether the four most recently opened documents will appear at the bottom of the File menu, where they can easily be retrieved. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

PrefEnvSettingsDlg

PrefSave

---

## **PrefMergeDelimit**

The PrefMergeDelimit command specifies the field and record delimiters for DOS delimited files to be used as secondary files in merges in WordPerfect. This command is the equivalent of setting the values in the Preferences Merge Settings dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefMergeDelimit command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefMergeDelimit (BeginField:*Character Expression*; EndField:*Character Expression*;  
BeginRecord:*Character Expression*; EndRecord:*Character Expression*)

### **parameters**

This command has four parameters:

#### **BeginField: EndField: BeginRecord: EndRecord:**

These character expressions specify the text and/or codes used to delimit the fields and records in text files. If you want the delimiters to contain tabs, line feeds, form feeds, or carriage returns, you must specify those codes in the Merge Preferences dialog box while recording a macro.

Each of these parameters inserts an "embedded" code into the character expression, which can be viewed using Reveal Codes when you edit the macro. If you want to insert an embedded code into an existing parameter in a macro, record another macro containing that code, open that macro, copy the embedded code to the Clipboard, then paste it into that parameter.

MergeDOSText

PrefMergeDlg

PrefSave

---

## **PrefMergeDlg**

The PrefMergeDlg command displays the Merge Preferences dialog box, which lets the user specify field and record delimiters. The PrefMergeDlg command is the equivalent of choosing Preferences from the File menu, then choosing Merge.

### **syntax**

PrefMergeDlg ( )

PrefMergeDelimit

---

## PrefPrintCopies

The PrefPrintCopies command specifies the default number of copies of each page to be printed, and whether WordPerfect or the printer will generate multiple copies. The PrefPrintCopies command is the equivalent of setting the options in the Multiple Copies group box in the Print Settings dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefPrintCopies command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefPrintCopies (NumberOfCopies:*Numeric Expression*; GeneratedBy:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **NumberOfCopies:**

This numeric expression specifies the default number of copies to print. This parameter should be a positive integer.

#### **GeneratedBy:**

This parameter specifies whether WordPerfect or the printer will handle printing multiple copies of each page. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

WordPerfect!	0
--------------	---

Printer!	1
----------	---

PrefPrintSettingsDlg

PrefSave

PrintOptions

---

## PrefPrintDoc

The PrefPrintDoc command specifies the default binding offset, graphics quality, and text quality for printed WordPerfect documents. This command is the equivalent of setting the Document Settings group box options in the Print Settings dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefPrintDoc command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefPrintDoc (*BindingOffset:Measurement Expression; GraphicsQuality:Value Set Member; TextQuality:Value Set Member*)

### **parameters**

This command has three parameters:

#### **BindingOffset:**

This measurement expression specifies the amount of extra space to allow along the binding edge of documents in WordPerfect.

#### **GraphicsQuality: TextQuality:**

These parameters specify the quality of printed graphics and text in WordPerfect documents. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

DoNotPrint!	0
Draft!	1
Medium!	2
High!	3

PrefPrintSettingsDlg

PrefSave

PrintOptions

---

## PrefPrintRedline

The PrefPrintRedline command specifies how and where redlined text will appear in WordPerfect documents. This command is the equivalent of setting Redline Method group box options in the Print Settings dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefPrintRedline command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefPrintRedline (Method:*Value Set Member*; Character:*Character Expression*)

### **parameters**

This command has two parameters:

#### **Method:**

This parameter specifies whether redlined text will be specific to your printer (on laser printers, for example, most redlined text is printed with a shaded background), or appear as a column of characters down the left or alternating sides of the page. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

PrinterDependent!	0
-------------------	---

Left!	1
-------	---

Alternating!	2
--------------	---

#### **Character:**

This character expression sets the character that appears on the left margin or alternating margins of redlined text. This character expression should be one character long. If the Method parameter is set to PrinterDependent!, the Character parameter is ignored and can be omitted.

PrefPrintSettingsDlg

PrefSave



---

## **PrefPrintSettingsDlg**

The PrefPrintSettingsDlg command displays the Print Settings dialog box, which lets the user specify percentages of normal font sizes for font size attributes, how many copies of each page to print, binding offset, graphics and text quality, and a redline method. The PrefPrintSettingsDlg command is the equivalent of choosing Preferences from the File menu, then choosing Print.

### **syntax**

PrefPrintSettingsDlg ( )

---

## PrefPrintSizeRatio

The PrefPrintSizeRatio command specifies a setting for each font attribute as a percentage of the current font. This command is the equivalent of setting Size Attribute Ratio group options in the Print Settings dialog box.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefPrintSizeRatio command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefPrintSizeRatio (*Fine:Numeric Expression; Small:Numeric Expression; Large:Numeric Expression; VeryLarge:Numeric Expression; ExtraLarge:Numeric Expression; SuperSubscript:Numeric Expression*)

### **parameters**

This command has six parameters:

#### **Fine: Small: Large: VeryLarge: ExtraLarge: SuperSubscript:**

These expressions set the font size as a percentage of the current font. You can also use a numeric expression to set the font size as a percentage of the current font. Each numeric expression should be a value from 1 to 100.

PrefPrintSettingsDlg

PrefSave

---

## PrefPrintWindowDriverSet

The PrefPrintWindowDriverSet command specifies whether Fast Graphics Printing is turned on. The Fast Graphics Printing option allows WordPerfect to print graphics substantially faster.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefPrintWindowDriverSet command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefPrintWindowDriverSet (FastGraphics:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **FastGraphics:**

This parameter specifies whether Fast Graphics Printing is turned on or off. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

PrefPrintSettingsDlg

PrefSave

---

## PrefRevealCodesColors

The PrefRevealCodesColors command sets a foreground and background color for a specified Reveal Codes item. This command is the equivalent of choosing a Reveal Codes Item, then selecting a color from the foreground palette and a color from the background palette.

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefRevealCodesColors command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefRevealCodesColors (FontAttribute:*Value Set Member*; Foreground:*Value Set Member*; BackGround:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **FontAttribute:**

This parameter specifies the Reveal Codes item to which the colors in the Foreground and Background parameters apply. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Text!	0
Codes!	1
Cursor!	2

#### **Foreground:**

This parameter specifies the foreground color for the item specified in the FontAttribute parameter. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

White!	0
DkGray!	1
LtRed!	2
LtYellow!	3
LtGreen!	4
LtCyan!	5
LtBlue!	6
LtMagenta!	7
LtGray!	8
Black!	9
DkRed!	10
DkYellow!	11
DkGreen!	12
DkCyan!	13

DkBlue!	14
DkMagenta!	15

**Background:**

This parameter specifies the background color for the item specified in the FontAttribute parameter. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

White!	0
DkGray!	1
DkCyan!	2
DkMagenta!	3
LtGray!	4
Black!	5
DkBlue!	6
LtRed!	7

PrefRevealCodesColorsDlg

PrefSave

---

## **PrefRevealCodesColorsDlg**

The PrefRevealCodesColorsDlg command displays the Reveal Codes Colors dialog box, which lets the user specify foreground and background colors for text, codes, and the insertion point when Reveal Codes is on. The PrefRevealCodesColorsDlg command is the equivalent of choosing Preferences from the File menu, then choosing Display, then choosing the Reveal Codes Colors button in the Display Settings dialog box.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PrefRevealCodesColorsDlg ( )

PrefRevealCodesColors

---

## PrefRuler

The PrefRuler command specifies the default Ruler display settings for WordPerfect. This command is the equivalent of setting Ruler options in the Environment Settings [dialog box](#).

If you want this preference option to remain set in the following sessions of WordPerfect, the PrefRuler command should be followed by a PrefSave command at some point in the macro. Otherwise, this preference option will revert to its former state when you quit WordPerfect.

### **syntax**

PrefRuler (TabsToGrid:*Value Set Member*; ShowRuler:*Value Set Member*;  
RulerButtonsOnTop:*Value Set Member*; AutoRulerDisplay:*Value Set Member*)

### **parameters**

This command has four parameters:

#### **TabsToGrid:**

This parameter specifies whether tabs on the ruler will "snap to" invisible gridlines that occur every 1/16th of an inch or 1/5th of a centimeter. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

No!	0
Yes!	1

#### **ShowRuler:**

This parameter specifies whether a ruler guide will be displayed when a tab stop is moved. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

No!	0
Yes!	1

#### **RulerButtonsOnTop:**

This parameter specifies whether ruler buttons should display on the top or the bottom of the ruler. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
---------------	---------------------------	-------------------

No!	0	RulerShow buttons are displayed on the bottom of the ruler.
Yes!	1	RulerShow buttons are displayed on the top of the ruler.

#### **AutoRulerDisplay:**

This parameter specifies whether the ruler will automatically display when a document is opened. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

No!	0
Yes!	1

PrefEnvSettingsDlg

PrefSave

RulerShow



---

## **PrefSave**

The PrefSave command saves changes that have been made to WordPerfect Preference options. Once this command is encountered, preference changes will apply to following sessions of WordPerfect. If a macro makes preference changes but does not save them with a PrefSave command, those changes will not apply to future sessions of WordPerfect.

### **syntax**

PrefSave ( )

---

## PrefToA

The PrefToA command specifies the default appearances of tables of authority in WordPerfect. This command is the equivalent of setting options in the ToA Preferences dialog box.

### **syntax**

PrefToA (DotLeaders:*Value Set Member*; Underlining:*Value Set Member*;  
BlankLineBetween:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **DotLeaders:**

This parameter specifies whether dot leaders should extend from the authority to its page number reference. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

#### **Underlining:**

This parameter specifies whether underlining will be allowed in the full form in tables of authority. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

NotAllowed!	0
-------------	---

Allowed!	1
----------	---

#### **BlankLineBetween:**

This parameter specifies whether there will be an extra hard return between each authority in tables of authority. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

ToADefine

---

## **PrefToADlg**

The PrefToADlg command displays the ToAPreferences dialog box, which lets the user specify whether generated tables of authorities will have dot leaders, underlining, and blank lines between authorities. The PrefToADlg command is the equivalent of choosing Preferences from the File menu, then choosing Table of Authorities.

### **syntax**

PrefToADlg ( )

---

## **PrefZoom**

This command allows you to specify the percent you would like to enlarge or reduce the viewing size of the displayed text and graphics.

### **syntax**

PrefZoom (Percent:*Numeric Expression*)

PrefZoomDlg

Zoom100

---

## **PrefZoomDlg**

The PrefZoomDlg command displays the Zoom Preferences dialog box and allows you to specify the amount you would like to enlarge or reduce the viewing size of displayed text and graphics during macro execution.

### **syntax**

PrefZoomDlg ( )

PrefZoom

Zoom100

---

## PrintDlg

The PrintDlg command displays the Print dialog box, which lets the user specify print options and document settings. The PrintDlg command is the equivalent of choosing Print from the File menu.

### **syntax**

PrintDlg ( )

PrinterInitialize

PrintDoc

PrintFull

PrintMultiplePages

PrintOptions

PrintPage

PrintSelected

---

## PrintDoc

The PrintDoc command prints a specified document that has already been saved on disk, without opening that document. This command lets you specify whether to print the entire document or selected pages. The PrintDoc command is the equivalent of specifying options in the Document On Disk [dialog box](#).

### syntax

PrintDoc (DocumentName:*Character Expression*; Range:*Character Expression*;  
OddEven:*Value Set Member*; DocumentSummary:*Value Set Member*; NoWarnings:*Value Set Member*)

### parameters

This command has five parameters:

#### DocumentName:

This character expression specifies the name or full [path](#) of the document to be printed. This parameter is the equivalent of specifying a name in the Filename [text box](#).

#### Range:

This character expression specifies the pages of the specified document to be printed. If this parameter is missing (the Range parameter is not used), WordPerfect assumes that the entire document should be printed. If this parameter is empty (the Range parameter is used, but no character expression is specified), WordPerfect checks the OddEven and DocumentSummary parameters and prints accordingly. If OddEven and DocumentSummary are also missing, then no pages are printed.

Note that this parameter is a character expression, not a numeric expression. This allows you to specify specific pages to be printed (for instance, if you want pages 13, 19, and 28 to be printed, this parameter would be "13, 19, 28"). For complete instructions on specifying a range, see Print Document on Disk in WordPerfect Reference.

#### OddEven:

This parameter specifies whether to print even pages, odd pages, or all pages of the specified document, corresponding to the Odd/Even Pages pop-up list in the Document on Disk dialog box. The valid values are:

Member	Numeric Equivalent
--------	--------------------

None!	0
Odd!	1
Even!	2
LogicalOdd!	3
LogicalEven!	4

#### DocumentSummary:

This parameter specifies whether the document summary will be printed. The valid values are:

Member	Numeric Equivalent	Definition
--------	--------------------	------------

No!	0	The document summary is not printed.
Yes!	1	The document summary is printed.

**NoWarnings:**

This parameter specifies whether to bypass message boxes that appear during command execution. When a message box is bypassed, the default option is automatically selected. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

PrintDlg



---

## PrinterCommand

Use PrinterCommand to insert a specified coded message into the active document at the insertion point. This command lets you take advantage of special functions your printer might provide. Using PrinterCommand is the equivalent of typing a command or filename in the Printer Command dialog box.

### **syntax**

PrinterCommand (CommandString:*Character Expression*; FileName:*Character String*)

### **parameters**

This command has two parameters:

#### **Command String:**

This character expression specifies the command to insert into the active document.

#### **FileName:**

This character string specifies the filename of a file that contains printer commands you want sent to the printer.

PrinterCommandDlg

---

## PrinterCommandDlg

The PrinterCommandDlg command displays the Printer Command dialog box, which lets the user specify a printer code to be inserted into the active document at the insertion point. This code lets the user access special functionality that the printer may have. The PrinterCommandDlg command is the equivalent of choosing Typesetting from the Layout menu, then choosing the Printer Command button in the Typesetting dialog box.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

PrinterCommandDlg ( )

PrinterCommand

---

## PrinterInitialize

The PrinterInitialize command sends information to your printer, so it can print documents using soft fonts you have stored on a drive. This process is called "downloading." The PrinterInitialize command is only useful if you are using a WordPerfect printer driver, not a Windows printer driver. The PrinterInitialize command is the equivalent of choosing Initialize Printer in the Print dialog box.

### **syntax**

PrinterInitialize ( )

PrintDlg

---

## PrinterSelect

The PrinterSelect command selects a printer driver (such as WordPerfect printer driver or Windows printer driver), a printer name, and a printer port. The PrinterSelect command is the equivalent of choosing Select Printer from the File menu and making the choices in the Select Printer dialog box for printer driver, printer, and port.

### **syntax**

PrinterSelect (PrinterType: *Value Set Member*; PrinterName: *Character Expression*;  
PrinterPort: *Character Expression*)

### **parameters**

This command has three parameters.

#### **PrinterType**

This parameter is a value set member. The PrinterType specifies whether you will be using WordPerfect printer drivers or Windows printer drivers.

#### **PrinterName**

This parameter is a character expression. The PrinterName is the name of the printer you use for printing. This name must be available when the associated printer driver is selected.

#### **PrinterPort**

This parameter is a character expression. The PrinterPort is the name of the port to which the printer is attached.

[PrinterSelectListCreate](#)

---

## PrinterSelectDlg

The PrinterSelectDlg command displays the Select Printer dialog box, which lets the user select a printer, delete a printer, update a printer, and so on. The PrinterSelectDlg command is the equivalent of choosing Select Printer from the File menu.

### **syntax**

PrinterSelectDlg ( )

---

## PrinterSelectListCreate

The PrinterSelectListCreate command creates a list of printers that are available with the printer driver specified in the PrinterType parameter.

### **syntax**

PrinterSelectListCreate (PrinterType: *Value Set Member*; PrinterCount)

### **parameters**

This command has two parameters.

#### **PrinterType**

This parameter is a value set member. The PrinterType specifies whether the list of printers available for selection will come from WordPerfect printer drivers or Windows printer drivers. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

WPPrinter!	0
------------	---

WindowsPrinter!	1
-----------------	---

#### **PrinterCount**

This parameter is a variable. The PrinterCount parameter contains the total from a count of all the printers listed in the list created by the PrinterSelectListCreate command.

[PrinterSelect](#)

[PrinterSelectListFree](#)

[PrinterSelectListItem](#)

---

## **PrinterSelectListFree**

The PrinterSelectListFree command frees the list of printers created with the PrinterSelectListCreate command.

### **syntax**

PrinterSelectListFree ( )

PrinterSelect

PrinterSelectListCreate

---

## PrinterSelectListItem

The PrinterSelectListItem command retrieves an item from the list of printers created with the PrinterSelectListCreate command.

### **syntax**

PrinterSelectListItem (Item: Numeric Expression; PrinterName; PrinterPort)

### **parameters**

This command has three parameters.

#### **Item**

This parameter is a numeric expression. When the PrinterSelectListCreate command creates a list of the printers available, the list is numbered. The Item specifies the number of the printer to be retrieved from the list that was created.

#### **PrinterName**

This parameter is a variable. The PrinterName variable will contain the name of the printer that was selected.

#### **PrinterPort**

This parameter is a variable. The PrinterPort variable will contain the name of the port to which the selected printer is attached.

[PrinterSelect](#)

[PrinterSelectListCreate](#)



---

## **PrintFull**

The PrintFull command prints the entire active document. The PrintFull command is the equivalent of choosing Full Document in the Print dialog box, then choosing Print.

### **syntax**

PrintFull ( )

PrintDlg

---

## PrintMultiplePages

The PrintMultiplePages command prints specified pages of the active document. This command is the equivalent of choosing the Multiple Pages option in the Print dialog box, then choosing Print, then filling in options in the Multiple Pages dialog box.

### **syntax**

PrintMultiplePages (Range:*Character Expression*; OddEven:*Value Set Member*;  
DocumentSummary:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **Range:**

This character expression specifies which pages in the current document to print. If this parameter is missing, WordPerfect assumes that the entire document should be printed. If this parameter is empty, WordPerfect checks the OddEven and DocumentSummary parameters and prints accordingly. If OddEven and DocumentSummary are also missing, then no pages are printed.

Note that this parameter is a character expression, not a numeric expression. This allows you to specify specific pages to be printed (for instance, if you want pages 13, 19, and 28 to be printed, this parameter would be "13, 19, 28"). For complete instructions on specifying a range, see Print Document in WordPerfect Reference.

#### **OddEven:**

This parameter specifies whether to print even pages, odd pages, or all pages of the document, corresponding to the Odd/Even Pages pop-up list in the Multiple Pages dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

None!	0
Odd!	1
Even!	2
LogicalOdd!	3
LogicalEven!	4

#### **DocumentSummary:**

This parameter specifies whether the document summary will be printed. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
---------------	---------------------------	-------------------

---

No!	0	The document summary is not printed
Yes!	1	The document summary is printed.

---

## PrintOptions

The PrintOptions command sets printing and document settings, including the binding offset, the printing quality, the number of copies to be printed, and the means of printing those copies. This command is the equivalent of setting the options in the Copies group box and Document Settings group box in the Print dialog box.

### **syntax**

PrintOptions (*Copies:Numeric Expression; CopiesBy:Value Set Member; Binding:Measurement Expression; TextQuality:Value Set Member; GraphicsQuality:Value Set Member*)

### **parameters**

This command has five parameters:

#### **Copies:**

This numeric expression specifies the number of copies to be printed. This numeric expression should be a positive integer. The valid range is 1-16383.

#### **CopiesBy:**

This parameter specifies whether WordPerfect or the printer will handle the printing of duplicate copies of the document. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

WordPerfect!	0
Printer!	1

#### **Binding:**

This measurement expression specifies the binding width for the document. If you do not want WordPerfect to add extra space to the margins to allow for binding, you may omit this parameter or set it to 0.

#### **Graphics Quality: Text Quality:**

These parameters specify the quality at which text or graphics will be printed. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

DoNotPrint!	0
Draft!	1
Medium!	2
High!	3

---

## **PrintPage**

The PrintPage command prints the current page in the active document. The "current page" is the page containing the insertion point. This command is the equivalent of choosing the Current Page option in the Print dialog box, then choosing Print.

### **syntax**

PrintPage ( )

PrintDlg

---

## **PrintPreview**

The PrintPreview command displays the document as it should appear when it prints. This command is the equivalent of choosing Print Preview from the File menu.

### **syntax**

PrintPreview ( )

---

## PrintSelected

The PrintSelected command prints any selected text. If no text is selected, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. The PrintSelected command is the equivalent of choosing the Selected Text option in the Print dialog box, then choosing Print.

### **syntax**

PrintSelected ( )

LABEL

ONERROR

PrintDlg

---

## **QUIT**

The QUIT command stops the macro. No commands following it are executed. If the macro currently running was started by another macro, both macros are stopped.

### **syntax**

QUIT

Show Me!

---

## REPEAT

REPEAT is the opening statement for the REPEAT-UNTIL conditional loop. When WordPerfect encounters a REPEAT command, it repeatedly plays all of the commands between the REPEAT command and the UNTIL command until the condition set in the UNTIL command is met. For example, consider the following commands:

```
ASSIGN (NUMBER; 1)
```

```
REPEAT
```

```
ASSIGN (NUMBER; NUMBER+1)
```

```
UNTIL (NUMBER=10)
```

In this macro, the third line adds 1 to the value of the variable NUMBER each time the loop repeats. When NUMBER has the value 10, the UNTIL command is true. At that time, the macro stops repeating the loop and goes to the command immediately below the UNTIL command.

Since the condition is evaluated at the end of the loop, a Repeat-Until loop will always execute at least once, regardless of whether the condition is true or false. For information on conditional loops whose test expressions are evaluated at the top of the loop, see WHILE and ENDWHILE.

### **syntax**

```
REPEAT
```

```
UNTIL
```

```
Show Me!
```



---

## RETURN

The RETURN command marks the end of a series of commands begun with a LABEL command. This command is usually used in conjunction with a CALL or CASE CALL command. When a RETURN command is encountered, macro execution returns to the command immediately following the command that called the LABEL command, and continues running from there.

If there is no CALL or CASE CALL to return to, and the macro containing this command is nested, this command signals the macro to return to the file from which it was nested. If the command is not in a nested file and there is no CALL or CASE CALL to return to, this command will stop the macro.

If a macro encounters a RETURN command before a CALL or CASE CALL command, the macro ends or returns to the macro that invoked the current macro.

### **syntax**

RETURN

CALL

CASE CALL

ONCANCEL CALL

ONERROR CALL

ONNOTFOUND CALL

RETURNCANCEL

RETURNERROR

RETURNNOTFOUND

RUN

Show Me!

---

## RETURNCANCEL

The RETURNCANCEL command marks the end of a series of commands begun with a LABEL command and is usually used in conjunction with a CALL or CASE CALL command. LABEL commands are called by other commands, so when a RETURNCANCEL command is encountered, macro execution returns to the command immediately following the command that called the LABEL command. (For example, a GO command calls a LABEL command. So when the RETURNCANCEL command is encountered, macro execution returns to the command following the GO command). At that point, the RETURNCANCEL command causes the macro to behave as if a cancel has occurred.

If an ONCANCEL command has been encountered earlier in the macro, macro execution will be transferred to the LABEL command specified in the ONCANCEL command. Otherwise, macro execution will stop.

If there is no CALL or CASE CALL to return to, and the macro containing this command is nested, this command signals the macro to return to the file from which it was nested. If the command is not in a nested file and there is no CALL or CASE CALL to return to, this command will stop the macro.

### **syntax**

RETURNCANCEL

CALL

CASECALL

LABEL

ONCANCEL

RETURN

---

## RETURNERROR

The RETURNERROR command marks the end of a series of commands begun with a LABEL command. This command is usually used in conjunction with a CALL or CASE CALL command. When a RETURNERROR command is encountered, macro execution returns to the command immediately following the command that called the LABEL command. (For example, a GO command calls a LABEL command. So when the RETURNERROR command is encountered, macro execution returns to the command following the GO command) At that point, the RETURNERROR command causes the macro to behave as if an error has occurred.

For example, if an ERROROFF command has been encountered, the macro will continue from that command. If an ONERROR command has been encountered earlier in the macro, macro execution will be transferred to the LABEL command specified in the ONERROR command. Otherwise, macro execution will stop.

If there is no CALL or CASE CALL to return to, and the macro containing this command is nested, this command signals the macro to return to the file from which it was nested, then to behave as if an error has occurred. If the command is not in a nested file and there is no CALL or CASE CALL to return to, this command will stop the macro.

### **syntax**

RETURNERROR

CALL

CASECALL

LABEL

ONERROR

RETURN

---

## RETURNNOTFOUND

The RETURNNOTFOUND command marks the end of a series of commands begun with a LABEL command. This command is usually used in conjunction with a CALL or CASE CALL command. When a RETURNNOTFOUND command is encountered, macro execution returns to the command immediately following the command that called the LABEL command. (For example, a GO command calls a LABEL command. So when the RETURNNOTFOUND command is encountered, macro execution returns to the command following the GO command) At that point, the RETURNNOTFOUND command causes the macro to behave as if a not found state exists.

For example, if an ONNOTFOUND command has been encountered earlier in the macro, macro execution will be transferred to the LABEL command specified in the ONNOTFOUND command. Otherwise, macro execution will stop.

### **syntax**

RETURNNOTFOUND

CALL

CASECALL

ONNOTFOUND

RETURN

---

## **RUN**

The RUN command runs the macro specified in the Macro parameter. After the macro runs, its "parent" macro (the macro that contains the RUN command) continues from the command immediately below the RUN command.

The macro this command runs must already be compiled. This means that you must have already played the macro once, or you must have compiled the macro using the Macro Facility.

### **syntax**

RUN (*Macro*)

### **parameters**

This command has one parameter:

#### **Macro**

This is a character expression. The expression must correspond to the name of a macro.

[CHAIN](#)

[RETURN](#)

[Show Me!](#)

---

## **Redisplay**

The Redisplay command rewrites the information on your computer screen without moving the insertion point in the text. This command is the equivalent of pressing Ctrl+F3.

### **syntax**

Redisplay ( )

---

## RevealCodes

Use the RevealCodes command to turn the Reveal Codes feature on or off. This command is similar to choosing Reveal Codes from the View menu.

### **syntax**

RevealCodes (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, this command will act as a "toggle."

### **State:**

This parameter specifies whether Reveal Codes is turned on or off. If Reveal Codes is already in the specified state, this command has no effect.

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

---

## RulerShow

Use the RulerShow command to show or hide the ruler. This command is the equivalent of choosing Ruler from the View menu.

### **syntax**

RulerShow (State:*Value Set Member*)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter specifies whether the ruler is to be displayed. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---



---

## **SPEED**

Use the SPEED command when you want the macro to run more slowly. The Tenth of a Second parameter represents how many tenths of a second will elapse between each macro command. For example, if you want the computer to wait one-half second between each macro command, use a SPEED (5) command. The maximum wait allowed between commands is 600 (one minute). The SPEED command is most useful for debugging macros and making "demonstration" macros.

To run the macro at full speed again, use a SPEED (0) command.

### **syntax**

SPEED (*Tenths of a Second*)

### **parameters**

This command has one parameter:

#### **Tenths of a Second**

This is a numeric expression between 0 and 600.

WAIT

Show Me!

---

## STRLEN

This command assigns the length (the number of characters) of the Character Expression parameter to the variable specified in the Variable parameter. For instance, a macro can find the length of the word "Mississippi" by using the STRLEN command as follows:

```
STRLEN (LN; "Mississippi")
```

This command assigns the length of the word "Mississippi" (11) to variable LN. (In the command above, "Mississippi" is in quotes so the macro will recognize it as a character expression.)

### **syntax**

STRLEN (*Variable; Character Expression*)

### **parameters**

This command has two parameters:

#### **Variable**

Any variable is a valid value for this parameter.

#### **Character Expression**

This parameter is a character expression.

[BYTELEN](#)

[SUBSTR](#)

[STRPOS](#)

[Show Me!](#)

---

## STRNUM

The STRNUM command converts the specified character expression into a number, and assigns it to the variable specified in the Variable parameter. The decimal point recognized by this command is that defined by the "sDecimal" setting in the "[intl]" section of the Windows WIN.INI file.

For example, the text 123 would be converted to the number 123. Although the two look the same on your screen, they are understood differently by the computer; you cannot add a number to a text string nor text to a numeric string. This command, along with the NUMSTR (Number to String) command, allows you to change one type of value to another.

### **syntax**

STRNUM (*Variable; Character Expression*)

### **parameters**

This command has two parameters:

#### **Variable**

Any variable is a valid value for this parameter.

#### **Character Expression**

This parameter is a character expression. It should contain a string of numeric characters. The decimal point character (.) is also allowable. If any other character (such as a letter, operator, or punctuation mark) is in the character expression, the character and all other characters following it are ignored. For instance, if a macro encounters the command STRNUM (PRICE; "679A1.321"), the variable PRICE would contain the number 679. The "A" and every character after it are ignored.

## NUMSTR

Show Me!

---

## **STRPOS**

The STRPOS command checks to see if the character expression specified in the Sub Expression parameter is included in the Original Expression parameter. If it is, the variable in the Variable parameter is assigned a number corresponding to the beginning position of the Sub Expression parameter in the Original Expression parameter.

For example, the command to find the location of the word "Perfect" in "WordPerfect" would be STRPOS (PLACE; "Perfect"; "WordPerfect"). Variable PLACE would get the value 5, because "Perfect" begins on the fifth character in "WordPerfect".

If the value of the Sub Expression parameter is not found in the Original Expression parameter, the variable in the Variable parameter is given the value 0.

### **syntax**

STRPOS (*Variable; Sub Expression; Original Expression*)

### **parameters**

This command has three parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Sub Expression**

This parameter is a character expression.

#### **Original Expression**

This parameter is a character expression.

BYTEPOS

STRLEN

SUBSTR

Show Me!

---

## SUBBYTE

The SUBBYTE command assigns part of an expression to a variable. The first byte assigned to the variable is the byte specified in the Start Position parameter. (WordPerfect counts the bytes in the Original Expression parameter from left to right.) The length of the expression assigned to the variable is specified in the Length parameter. This command works in bytes, not characters. If you want to count your strings in characters, use the SUBSTR command.

For example, if a macro has the command SUBBYTE (MIDDLE; 5; 3; "Colorado"), the Start Position Expression parameter is 5, so the SUBBYTE command begins the substring at the fifth byte (for standard characters, each character is one byte long). From that place the Length parameter counts over three bytes ("rad"). Those three characters are assigned to variable MIDDLE.

### **syntax**

SUBBYTE (*Variable; Start Position; Length; Original Expression*)

### **parameters**

This command has four parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Start Position**

This numeric expression specifies the first byte to be used from the Original Expression parameter.

#### **Length**

This numeric expression specifies how many bytes in the Original Expression parameter will be used.

#### **Original Expression**

This parameter is a character expression. SUBBYTE assigns part of this expression to the variable specified in the Variable parameter.

BYTELEN

BYTEPOS

Show Me!

---

## **SUBSTR**

The SUBSTR command assigns part of an expression to a variable. The first character assigned to the variable is the character specified in the Start Position parameter. (WordPerfect counts the characters in the Original Expression parameter from left to right.) The length of the expression assigned to the variable is specified in the Length parameter.

For example, if a macro contains the command SUBSTR (MIDDLE; 5; 3; "Colorado"), the variable MIDDLE would get the value "rad". The Start Position Expression parameter is 5, so the substring starts at the fifth character. From that character ("r"), the Length parameter counts over three characters. Those three characters are assigned to variable MIDDLE.

### **syntax**

SUBSTR (*Variable; Start Position; Length; Original Expression*)

### **parameters**

This command has four parameters:

#### **Variable**

Any variable is valid for this parameter.

#### **Start Position**

This numeric expression specifies the first character to be used from the Original Expression parameter.

#### **Length**

This numeric expression specifies how many characters in the Original Expression parameter will be used.

#### **Original Expression**

This parameter is a character expression. SUBSTR assigns part of this expression to the variable specified in the Variable parameter.

STRLEN

STRPOS

Show Me!

---

## ScrollScreenLeft

Use the ScrollScreenLeft command to make the left part of the document window visible. This command is the equivalent of pressing Ctrl+PgUp or clicking the left arrow in the horizontal scroll bar.

### **syntax**

ScrollScreenLeft ( )

ScrollScreenRight

---

## ScrollScreenRight

Use the ScrollScreenRight command to make the right part of the document window visible. This command is the equivalent of pressing Ctrl+PgDn or clicking the right arrow in the horizontal scroll bar.

### **syntax**

ScrollScreenRight ( )

ScrollScreenLeft



---

## SearchDlg

The SearchDlg command displays the Search dialog box, which lets the user specify text and codes to search for. The SearchDlg command is the equivalent of choosing Search from the Edit menu.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SearchDlg ( )

---

## SearchNext

The SearchNext command searches forward from the insertion point for the search text most recently specified in a SearchText product command or in the Search or Search and Replace dialog boxes. This command is the equivalent of choosing Search Next from the Edit menu.

### **syntax**

SearchNext ( )

SearchDlg

SearchPrevious

SearchText

---

## SearchPrevious

The SearchPrevious command searches backward from the insertion point for the search text specified in a SearchText product command or in the Search or Search and Replace dialog boxes. This command is the equivalent of choosing Search Previous from the Edit menu.

### **syntax**

SearchPrevious ( )

SearchDlg

SearchNext

SearchText

---

## SearchReplace

The SearchReplace command replaces occurrences of one specified string of text and codes with another. You can use this command to search and replace forward or backward, through the entire document or just through the main body of the document. You can also specify whether the first occurrence or every occurrence found should be replaced. The SearchReplace command is the equivalent of setting options in the Search and Replace dialog box, then choosing Replace or Replace All.

### **syntax**

SearchReplace (SearchString:*Character Expression*; SearchDirection:*Value Set Member*; ReplacementScope:*Value Set Member*; ReplacementString:*Character Expression*; ReplacementAction:*Value Set Member*)

### **parameters**

This command has five parameters:

#### **SearchString:**

This character expression specifies the text and codes to be replaced with the information in the ReplacementString parameter. If you want the macro to search for codes, you must specify those codes in the Codes dialog box while recording a macro. This inserts an "embedded" code into the SearchString character expression, which can be viewed using Reveal Codes. If you want to insert an embedded code into an existing SearchString parameter in a macro, record another macro containing that code, open that macro, turn on Reveal Codes to find the embedded code, copy the code into the Clipboard, then paste it into the expression to be used as the SearchString parameter.

#### **SearchDirection:**

This parameter has two values, corresponding to the Direction pop-up list in the Search and Replace dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Forward!	0
----------	---

Backward!	1
-----------	---

#### **ReplacementScope:**

This parameter has two values corresponding to the Search Document Body Only check box in the Search and Replace dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Extended!	0
-----------	---

DocOnly!!	1
-----------	---

Extended! means that the search will include footnotes, endnotes, headers, text boxes, and so on.

#### **ReplacementString:**

This character expression specifies the text that should replace the information in the SearchString parameter. If you want the replacement text to contain codes, you must specify those codes in the Codes dialog box while recording a macro. This inserts an "embedded" code into the ReplacementString character expression, which can be viewed using Reveal Codes. If you want to insert an embedded code into an existing ReplacementString

parameter in a macro, record another macro containing that code, open that macro, copy the embedded code to the Clipboard, then paste it into the expression to be used as the ReplacementString parameter.

**ReplacementAction:**

This parameter specifies whether the first instance or every instance of the text and codes in the SearchString parameter should be replaced. This parameter is the equivalent of choosing Replace or Replace All in the Search and Replace dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

ReplaceAll!	1
-------------	---

ReplaceOne	2
------------	---

SearchDlg

SearchReplaceDlg

SearchText

---

## SearchReplaceDlg

The SearchReplaceDlg command displays the Search and Replace dialog box, which finds occurrences of a specified string of text and codes and lets the user replace the string with another string. The SearchReplaceDlg command is the equivalent of choosing Replace from the Edit menu.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SearchReplaceDlg ( )

SearchDlg

---

## SearchText

The SearchText command searches for a specified string of text and codes. You can use this command to search forward or backward through the entire document or through parts of the document. The SearchText command is the equivalent of setting options in the Search dialog box, then choosing Search.

### **syntax**

SearchText (SearchString:*Character Expression*; SearchDirection:*Value Set Member*;  
SearchScope:*Value Set Member*)

### **parameters**

This command has three parameters:

#### **SearchString:**

This character expression specifies the text and codes to search for. If you want the macro to search for codes, you must specify those codes in the Codes dialog box while recording your macro. Doing so will insert an "embedded" code into the SearchString character expression, which can then be viewed using Reveal Codes when you edit the macro. If you want to insert an embedded code into an existing SearchString parameter in a macro, record another macro containing that code, open that macro, turn on Reveal Codes to find the embedded code, copy the code to the Clipboard, then paste it into the expression to be used as the SearchString parameter.

#### **SearchDirection:**

This parameter has two values, corresponding to the Direction pop-up list in the Search dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Forward!	0
Backward!	1

#### **SearchScope:**

This parameter has three values. Two values correspond to the Search Document Body Only check box in the Search dialog box. The other value, WithinBlock!, is the equivalent of pressing Search Next in the Search and Replace dialog box while a block of text is selected. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Description</b>
Extended!	0	Search footnotes, endnotes, headers, and text boxes.
DocOnly!	1	Search only the main body of the document.
WithinBlock!	2	Search a block of selected text.

### SearchReplace

---

## SelectAll

The SelectAll command selects all text and graphics in the active window. For example, if a footer is currently being created when a macro encounters this command, the entire contents of the footer is selected. If a document window is active when a macro encounters this command, everything in the document is selected.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectAll ( )



---

## SelectCell

Use the SelectCell command in tables to select the cell at the insertion point (this is not the same as selecting the text in a cell; the SelectCell command selects all cell formatting and attributes as well as the text). Attributes given to this selection will supersede any other attributes in the document. To extend a selection started by the SelectCell command, use commands beginning with "Sel", such as SelectCharNext, SelectLineBegin, SelectLineDown, and so forth.

### **syntax**

SelectCell ( )

---

## SelectCellDown

The SelectCellDown command moves the insertion point down to the next row in the table, extending the selection from the anchor point to the new location. If the insertion point is not in a table, this command terminates the macro or redirects macro execution to the LABEL command specified by an ONERROR command. The SelectCellDown command is the equivalent of pressing Alt+Shift+Down Arrow. The only difference between this command and the SelectCellDownArrow command is that this command will not move the insertion point below the table when the insertion point is in the bottom row of a table.

### **syntax**

SelectCellDown ( )

LABEL

ONERROR

SelectCellUp

SelectCellDownArrow

---

## SelectCellDownArrow

The SelectCellDownArrow command moves the insertion point down to the next row in the table, extending the selection from the anchor point to the new location. This command has no effect outside a table. The only difference between this command and the SelectCellDown command is that this command will move the insertion point below the table when the insertion point is in the bottom row of a table.

This command is not recordable. To use the SelectCellDownArrow command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Shift+Down Arrow, the macro facility records the SelectLineDown ( ) command.

### **syntax**

SelectCellDownArrow ( )

SelectCellUp

SelectCellDown

---

## SelectCellLeft

The SelectCellLeft command moves the insertion point left to the previous column in the table, extending the selection from the anchor point to the new location. If the insertion point is at the leftmost column in the table, the insertion point is moved to the rightmost column in the table, one row up. If the insertion point is outside of a table, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command.

This command is not recordable. To use the SelectCellLeft command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Alt+Shift+Left Arrow, the macro facility records the SelectColumnPrevious ( ) command.

### **syntax**

SelectCellLeft ( )

LABEL

ONERROR

SelectCellRight

PosCellPrevious

---

## SelectCellRight

The SelectCellRight command moves the insertion point right to the next column in the table, extending the selection from the anchor point to the new location. If the insertion point is at the rightmost column in the table, the insertion point is moved to the leftmost column in the table, one row down. If the insertion point is outside of a table, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command.

This command is not recordable. To use the SelectCellRight command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Alt+Shift+Right Arrow, the macro facility records the SelectColumnNext ( ) command when you execute those keystrokes.

### **syntax**

SelectCellRight ( )

LABEL

ONERROR

SelectCellLeft

PosCellNext

---

## SelectCellUp

The SelectCellUp command moves the insertion point up to the next row in the table, extending the selection from the anchor point to the new location. If the insertion point is not in a table, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. The SelectCellUp command is the equivalent of pressing Alt+Shift+Up Arrow. The only difference between this command and the SelectCellUpArrow command is that this command will not move the insertion point above the table when the insertion point is in the top row of a table.

### **syntax**

SelectCellUp ( )

LABEL

ONERROR

SelectCellDown

SelectCellUpArrow

---

## SelectCellUpArrow

The SelectCellUpArrow command moves the insertion point up to the next row in the table, extending the selection from the anchor point to the new location. This command is invalid outside of a table. The only difference between this command and the SelectCellUp command is that this command will move the insertion point above the table when the insertion point is in the top row of a table.

This command is not recordable. To use the SelectCellUpArrow command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Shift+Up Arrow, when you do so while recording a macro, the macro facility records the SelectLineUp ( ) command.

Note also that when there is text in the cell, Shift+Up Arrow will select one line of the text; while SelectCellUpArrow will still extend the selection to the cell in the row above.

### **syntax**

SelectCellUpArrow ( )

SelectCellDownArrow

SelectCellUp

---

## SelectCharNext

The SelectCharNext command moves the insertion point one character to the right, and selects all codes and text between the previous position of the insertion point and the new position. If the insertion point is at the end of the document, this command has no effect. This command is the equivalent of pressing Shift+Right Arrow.

### **syntax**

SelectCharNext ( )

SelectCharPrevious



---

## SelectCharPrevious

The SelectCharPrevious command moves the insertion point one character to the left, and selects all codes and text between the previous position of the insertion point and the new position. If the insertion point is at the very beginning of the document, this command has no effect. This command is the equivalent of pressing Shift+ Left Arrow.

### **syntax**

SelectCharPrevious ( )

SelectCharNext

---

## SelectColumn

The SelectColumn command selects the contents of the column containing selected text. If there is no selected text, this command has no effect. This command is the equivalent of choosing Select from the Edit menu, then choosing Tabular Column.

### **syntax**

SelectColumn ( )

---

## SelectColumnBottom

The SelectColumnBottom command moves the insertion point to the bottom of the current column, and selects all codes and text between the previous position of the insertion point and the new position. This command selects text even if columns are not on. This command is the equivalent of pressing Shift+Alt+End in a column.

### **syntax**

SelectColumnBottom ( )

SelectColumnTop

---

## SelectColumnNext

The SelectColumnNext command moves the insertion point one column to the right, and selects all codes and text between the previous position of the insertion point and the new position. If the insertion point is in the rightmost column, this command moves the insertion point to the leftmost column, one line down. This command selects text even if columns are not on. This command is the equivalent of pressing Shift+Alt+Right Arrow in columns.

### **syntax**

SelectColumnNext ( )

SelectColumnPrevious

---

## SelectColumnPrevious

The SelectColumnPrevious command moves the insertion point one column to the left, and selects all codes and text between the previous position of the insertion point and the new position. If the insertion point is in the leftmost column, this command moves the insertion point to the rightmost column, one line up. This command selects text even if columns are not on. This command is the equivalent of pressing Shift+Alt+Left Arrow in columns.

### **syntax**

SelectColumnPrevious ( )

SelectColumnNext

---

## SelectColumnTop

The SelectColumnTop command moves the insertion point to the top of the current column, and selects all codes and text between the previous position of the insertion point and the new position. This command selects text even if columns are not on.

This command is the equivalent of pressing Shift+Alt+Home in a column.

### **syntax**

SelectColumnTop ( )

SelectColumnBottom

---

## SelectDocBottom

The SelectDocBottom command moves the insertion point to the end of the active document, and selects all codes and text between the previous position of the insertion point and the new position. This command is the equivalent of pressing Shift+Ctrl+End.

### **syntax**

SelectDocBottom ( )

SelectDocTop

---

## SelectDocTop

The SelectDocTop command moves the insertion point to the beginning of text in the active document, and selects all codes and text between the previous position of the insertion point and the new position. This command is the equivalent of pressing Shift+Ctrl+Home.

### **syntax**

SelectDocTop ( )

SelectDocBottom

SelectDocVeryTop



---

## SelectDocVeryTop

The SelectDocVeryTop command moves the insertion point to the very beginning of the active document (before all codes and text), and selects all codes and text between the previous position of the insertion point and the new position.

This command is not recordable. To use the SelectDocVeryTop command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Ctrl+Shift+Home, Ctrl+Shift+Home, the macro facility records the SelectDocTop ( ) command twice.

### **syntax**

SelectDocVeryTop ( )

PosDocVeryTop

SelectDocBottom

SelectDocTop

---

## SelectLineBegin

The SelectLineBegin command moves the insertion point to the beginning of the current line, and selects all codes and text between the previous position of the insertion point and the new position. This command is the equivalent of pressing Shift+Home.

### **syntax**

SelectLineBegin ( )

SelectLineEnd

---

## SelectLineDown

The SelectLineDown command moves the insertion point one line down, and selects all codes and text between the previous position of the insertion point and the new position. If the insertion point is already at the last line in the document, this command has no effect. The SelectLineDown command is the equivalent of pressing Shift+Down Arrow in a document window.

### **syntax**

SelectLineDown ( )

SelectLineUp

---

## SelectLineEnd

The SelectLineEnd command moves the insertion point to the end of the current line, and selects all codes and text between the previous position of the insertion point and the new position. This command is the equivalent of pressing Shift+End.

### **syntax**

SelectLineEnd ( )

SelectLineBegin

---

## SelectLineUp

The SelectLineUp command moves the insertion point one line up, and selects all codes and text between the previous position of the insertion point and the new position. If the insertion point is already at the first line in the document, this command has no effect. The SelectLineUp command is the equivalent of pressing Shift+Up Arrow in a document window.

### **syntax**

SelectLineUp ( )

SelectLineDown

---

## SelectLineVeryBegin

The SelectLineVeryBegin command moves the insertion point to the beginning of the current line, before all codes and text, and selects all codes and text between the previous position of the insertion point and the new position.

This command is not recordable. To use the SelectLineVeryBegin command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Shift+Home,Shift+Home, the macro facility records the SelectLineBegin ( ) command twice.

### **syntax**

SelectLineVeryBegin ( )

PosLineVeryBegin

PosLineVeryEnd

SelectLineBegin

SelectLineVeryEnd

---

## SelectLineVeryEnd

The SelectLineVeryEnd command moves the insertion point to the end of the current line, after all codes and text, selecting from the previous location to the new location of the insertion point.

This command is not recordable. To use the SelectLineVeryEnd command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Shift+End,Shift+End, the macro facility records the SelectLineEnd ( ) command twice.

### **syntax**

SelectLineVeryEnd ( )

PosLineVeryBegin

PosLineVeryEnd

SelectLineBegin

SelectLineVeryBegin

---

## SelectMode

Use the SelectMode command to turn the Select Mode feature on or off. The Select Mode feature causes WordPerfect to use the WordPerfect 5.1 for DOS method of selecting text, rather than the standard Windows method. This means that regular positioning keys extend a selection, rather than cancelling it, and if you press a character, the selection is extended to include that character, rather than replacing the selection with that character. The SelectMode command is similar to pressing F8.

### **syntax**

SelectMode (State:Value Set Member)

### **parameters**

This command has one parameter. If the parameter is left out, the command acts as a "toggle."

### **State:**

This parameter specifies whether Select Mode is to be turned on or off. The members and corresponding numeric equivalents of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
On!	1



---

## SelectPage

The SelectPage command selects the contents of the page containing the insertion point.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectPage ( )

---

## SelectPageNext

The SelectPageNext command moves the insertion point to the beginning of the following page in the active document, and selects all codes and text from the previous position of the insertion point and its new position. If the insertion point is already in the last page, the SelectPageNext command moves the insertion point to the end of the document. This command is the equivalent of pressing Shift+Alt+PgDn.

### **syntax**

SelectPageNext ( )

SelectPagePrevious

---

## SelectPagePrevious

The SelectPagePrevious command moves the insertion point to the beginning of the previous page in the active document, and selects all codes and text between the previous position of the insertion point and its new position. If the insertion point is already in the first page, this command has no effect. The SelectPagePrevious command is the equivalent of pressing Shift+Alt+PgUp.

### **syntax**

SelectPagePrevious ( )

SelectPageNext

---

## SelectParagraph

The SelectParagraph command selects the entire paragraph containing the insertion point. This command is the equivalent of clicking the mouse quickly four times in a paragraph or choosing Select from the Edit menu, then choosing Paragraph.

### **syntax**

SelectParagraph ( )

---

## SelectParagraphNext

The SelectParagraphNext command moves the insertion point to the beginning of the next paragraph, and selects all codes and text between the previous position of the insertion point and its new position. If the insertion point is in the last paragraph in the document, this command moves the insertion point to the end of the document. The SelectParagraphNext command is the equivalent of pressing Shift+Ctrl+Down Arrow.

### **syntax**

SelectParagraphNext ( )

SelectParagraphPrevious

---

## SelectParagraphPrevious

The SelectParagraphPrevious command moves the insertion point to the beginning of the current paragraph, and selects all codes and text between the previous position of the insertion point and its new position. If the insertion point is at the beginning of a paragraph, this command moves the insertion point to the beginning of the preceding paragraph. The SelectParagraphPrevious command is the equivalent of pressing Shift+Ctrl+Up Arrow.

### **syntax**

SelectParagraphPrevious ( )

SelectParagraphNext

---

## SelectRectangle

The SelectRectangle command creates a rectangular selection of text, using the beginning and ending points of an existing selection as, respectively, upper-left and lower-right corners of that rectangle. This command is the equivalent of choosing Select from the Edit menu, then choosing Rectangle.

### **syntax**

SelectRectangle ( )

---

## SelectScreenDown

The SelectScreenDown command selects from the insertion point to the bottom of the screen the first time it is used, then down one screen at a time every time it is used immediately thereafter. This command is the equivalent of pressing Shift+PgDn.

### **syntax**

SelectScreenDown ( )

SelectScreenUp



---

## SelectScreenLeft

The SelectScreenLeft command moves the insertion point to the left side of the screen the first time it is used, then left one screen at a time (or to the left margin of the document) every time it is used immediately thereafter. This command is the equivalent of pressing Shift+Home+Left Arrow.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectScreenLeft ( )

PosScreenLeft

SelectScreenRight

---

## SelectScreenRight

The SelectScreenRight command moves the insertion point to the right side of the screen the first time it is used, then right one screen at a time (or to the right margin of the document) every time it is used immediately thereafter. This command is the equivalent of pressing Shift+Home+Right Arrow.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectScreenRight ( )

PosScreenRight

SelectScreenLeft

---

## SelectScreenUp

The SelectScreenUp command selects from the insertion point to the top of the screen the first time it is used, then up one screen at a time every time it is used immediately thereafter. This command is the equivalent of pressing Shift+PgUp.

### **syntax**

SelectScreenUp ( )

SelectScreenDown

---

## SelectSentence

The SelectSentence command selects the sentence containing the insertion point. This command is the equivalent of triple-clicking the mouse pointer on a sentence or choosing Select from the Edit menu, then choosing Sentence.

### **syntax**

SelectSentence ( )

---

## SelectSentenceNext

The SelectSentenceNext command selects everything from the insertion point to the beginning of the next sentence. If text is already selected, the command extends the selection from the anchor point to the beginning of the next sentence. This command is the equivalent of triple-clicking the mouse to the right of the cursor while holding down Shift. This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectSentenceNext ( )

SelectSentencePrevious

---

## SelectSentencePrevious

The SelectSentencePrevious command selects everything from the insertion point to the end of the previous sentence. This command is the equivalent of triple-clicking the mouse to the left of the cursor while holding down Shift. This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectSentencePrevious ( )

SelectSentenceNext

---

## SelectTable

The SelectTable command selects all cells in the table containing the insertion point. This command is the equivalent of triple-clicking the mouse when the insertion point becomes a selection arrow inside a table. (The insertion point becomes an arrow when it is near the left or top border of a cell.)

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectTable ( )

SelectTableColumn

SelectTableRow

---

## SelectTableColumn

The SelectTableColumn command selects all cells in the current column of the current table. This command is the equivalent of double-clicking the mouse when the insertion point becomes an up selection arrow inside a table. (The insertion point becomes an up selection arrow when it is near the top border of a cell.)

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectTableColumn ( )

SelectTable

SelectTableRow



---

## SelectTableColumnExtendLeft

Use the SelectTableColumnExtendLeft command to select cells from the anchor cell to leftmost cell. All rows that contain any selected cells are affected.

This command is not recordable. To use the SelectTableColumnExtendLeft command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Shift+Home in a table, the macro facility records the SelectLineBegin ( ) command when you execute these keystrokes if at least one cell is selected,.

### **syntax**

SelectTableColumnExtendLeft ( )

SelectTableColumnExtendRight

---

## SelectTableColumnExtendRight

Use the SelectTableColumnExtendRight command to select cells from the anchor cell to the rightmost cell. All rows currently containing any selected cells are affected.

This command is not recordable. To use the SelectTableColumnExtendRight command, you must add it to the macro by editing the macro. Although this command is the equivalent of pressing Shift+End in a table, the macro facility records the SelectLineEnd ( ) command when these keystrokes are executed if at least one cell is selected.

### **syntax**

SelectTableColumnExtendRight ( )

SelectTableColumnExtendLeft

---

## SelectTableRow

The SelectTableRow command selects all of the cells in the current row of the current table. This command is the equivalent of double-clicking the mouse when the insertion point becomes a left selection arrow inside a table. (The insertion point becomes a left selection arrow when it is near the left border of a cell.)

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectTableRow ( )

SelectTable

SelectTableColumn

---

## SelectWord

The SelectWord command selects the word containing the insertion point. This command is the equivalent of double-clicking the mouse pointer on a word.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

SelectWord ( )

---

## SelectWordNext

The SelectWordNext command selects everything from the insertion point to the beginning of the next word. This command is the equivalent of pressing Shift+Ctrl+Right Arrow.

### **syntax**

SelectWordNext ( )

---

## SelectWordPrevious

The SelectWordPrevious command selects everything from the insertion point to the end of the previous word. This command is the equivalent of pressing Shift+Ctrl+Left Arrow.

### **syntax**

SelectWordPrevious ( )

---

## ShortMenus

The ShortMenus command specifies whether WordPerfect displays short menus or full menus. Short menus show only those items you will probably use the most, while full menus show all possible menu features. This command is the equivalent of choosing Short Menus from the View menu. (*Note:* Short Menus will not be available from the View menu unless you enable the Short Menu option. In order to enable Short Menus, you must put EnableShortMenus=1 into your WPWP\*.INI file.)

### **syntax**

ShortMenus (State:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **State:**

This parameter specifies whether short or long (full) menus will display in WordPerfect. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Short!	0
Long!	1

---

## Sort

Use the Sort command to sort lines of text, paragraphs, merge records, or rows in a table in the active document or in selected text. This command is the equivalent of setting options in the Sort dialog box, then choosing OK.

### **syntax**

Sort (RecordType:Value Set Member; SortOrder:Value Set Member; Selection:Character Expression; Key1Type:Value Set Member; Key1Level1:Numeric Expression; Key1Level2:Numeric Expression; Key1Level3:Numeric Expression; Key2Type:Value Set Member; Key2Level1:Numeric Expression; Key2Level2:Numeric Expression; Key2Level3:Numeric Expression; Key3Type:Value Set Member; Key3Level1:Numeric Expression; Key3Level2:Numeric Expression; Key3Level3:Numeric Expression; Key4Type:Value Set Member; Key4Level1:Numeric Expression; Key4Level2:Numeric Expression; Key4Level3:Numeric Expression; Key5Type:Value Set Member; Key5Level1:Numeric Expression; Key5Level2:Numeric Expression; Key5Level3:Numeric Expression; Key6Type:Value Set Member; Key6Level1:Numeric Expression; Key6Level2:Numeric Expression; Key6Level3:Numeric Expression; Key7Type:Value Set Member; Key7Level1:Numeric Expression; Key7Level2:Numeric Expression; Key7Level3:Numeric Expression; Key8Type:Value Set Member; Key8Level1:Numeric Expression; Key8Level2:Numeric Expression; Key8Level3:Numeric Expression; Key9Type:Value Set Member; Key9Level1:Numeric Expression; Key9Level2:Numeric Expression; Key9Level3:Numeric Expression)

### **parameters**

This command has 39 parameters:

#### **RecordType:**

This parameter specifies whether WordPerfect will sort the lines, paragraphs, merge records, or table rows in the document or in selected text. It corresponds to the options in the Record Type group box in the Sort dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Line!	1
Paragraph!	2
MergeRec!	3
TableRow!	4

#### **SortOrder:**

This parameter specifies whether the document or selected text will be sorted in ascending order, descending order, or not at all. The values correspond to the options in the Sort Order group box in the Sort dialog box. The default is Ascending!. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Ascending!	1
Descending!	2
NoSort!	3

#### **Selection:**

This character expression specifies criteria used to extract records.



**Key1Type: Key2Type: Key3Type: Key4Type: Key5Type: Key6Type: Key7Type:  
Key8Type: Key9Type:**

For each of these parameters you specify the sort key and follow it with the type of key it will be interpreted as; a number (Numeric!) or an alphanumeric word (Alpha!). Alphanumeric words are compared in the traditional alphabetic style: character-by-character to determine which should precede the other. Numbers are compared by the value of the number.

This parameter is the first in the repeating sequence of Key xType; followed by the parameter listed below: Key xLevel1; Key xLevel2; Key xLevel3 (where x is the key number). This sequence needs to be repeated only for the number of keys by which you want to sort. The default is Alpha!. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Numeric!	0
Alpha!	1

**KeyxLevelx:**

Key1Level1: Key2Level1: Key3Level1: Key4Level1: Key5Level1: Key6Level1: Key7Level1:  
Key8Level1: Key9Level1: Key1Level2: Key2Level2: Key3Level2: Key4Level2: Key5Level2:  
Key6Level2: Key7Level2: Key8Level2: Key9Level2: Key1Level3: Key2Level3: Key3Level3:  
Key4Level3: Key5Level3: Key6Level3: Key7Level3: Key8Level3: Key9Level3:

These numeric expressions specify the position of up to nine "key" words in each record. (A record's keys determine its position in the sorted document.) A key is specified by dividing a record into "levels." Each level is defined according to the current record type. The following table describes what the KeyLevel parameters represent for each record type:

<b>Record Type</b>	<b>Level 1</b>	<b>Level 2</b>	<b>Level 3</b>
--------------------	----------------	----------------	----------------

Line	Field*	not used	Word
Paragraph	Line	Field*	Word
MergeRec	Field**	Line	Word
TableRow	Cell	Line	Word

\*A tab separates two fields.

\*\*An {END FIELD} (or ^E) and line break separate two fields.

If a Level1 parameter is missing, the key which corresponds to that level as defined in this parameter is not used for sorting. If another level parameter is missing, the value "1" is assumed. All KeyLevel parameter values should be integers. Level1 parameters should have only positive values. Level2 and Level3 parameters may have either positive or negative values.

SortDlg

---

## SortDlg

The SortDlg command displays the Sort dialog box, which lets the user sort selected text or the entire active document. The SortDlg command is the equivalent of choosing Sort from the Tools menu.

### **syntax**

SortDlg ( )

Sort

---

## Speller

The Speller command starts the Speller program, which lets the user spell-check the active document. The Speller command is the equivalent of choosing Speller from the Tools menu.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

Speller ( )

Grammatik

Thesaurus

---

## SpreadsheetImportDlg

The SpreadsheetImportDlg command displays the Import Spreadsheet dialog box, which lets the user specify the information necessary to retrieve spreadsheet data into WordPerfect. This command is the equivalent of choosing Spreadsheet from the Tools menu, then choosing Import.

### **syntax**

SpreadsheetImportDlg ( )

SpreadsheetImportLink

---

# SpreadsheetImportLink

Use the SpreadsheetImportLink command to create or edit a spreadsheet link to the current document, or to retrieve spreadsheet information into the current document. The SpreadsheetImportLink command is the equivalent of setting options in the Create Spreadsheet Link, Edit Spreadsheet Link, or Import Spreadsheet dialog box.

## **syntax**

SpreadsheetImportLink (Operation:*Value Set Member*; Filename:*Character Expression*; RangeName:*Character Expression*; Type:*Value Set Member*)

## **parameters**

This command has four parameters:

### **Operation:**

This parameter specifies whether the command should create a spreadsheet link, edit an existing link, or import spreadsheet information. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Import!	0
Create!	1
Edit!	2

### **Filename:**

This character expression specifies the name or full path of the spreadsheet file to be linked to (or imported into) the active document. This parameter is the equivalent of typing in the Filename box in the Create Spreadsheet Link, Edit Spreadsheet Link, or Import Spreadsheet dialog box.

### **RangeName:**

This character expression specifies the range of cells to import from the spreadsheet file. This parameter is the equivalent of typing a value for the Range text box in the Create Spreadsheet Link, Edit Spreadsheet Link, or Import Spreadsheet dialog box.

### **Type:**

This parameter specifies whether the spreadsheet information to be linked or imported should be in a text or table format. This parameter is the equivalent of choosing Table or Text in the Create Spreadsheet Link, Edit Spreadsheet Link, or Import Spreadsheet dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Text!	0
Table!	1

SpreadsheetImportDlg

SpreadsheetLinkCreateDlg

SpreadsheetLinkEditDlg

---

## SpreadsheetLinkCreateDlg

The SpreadsheetLinkCreateDlg command displays the Create Spreadsheet Link dialog box, which lets the user specify the information necessary to link a spreadsheet file with the active document in WordPerfect. This command is the equivalent of choosing Spreadsheet from the Tools menu, then choosing Create Link.

### **syntax**

SpreadsheetLinkCreateDlg ( )

SpreadsheetLinkEditDlg

SpreadsheetImportLink

---

## SpreadsheetLinkEditDlg

The SpreadsheetLinkEditDlg command displays the Edit Spreadsheet Link dialog box, which lets the user edit an existing link between a spreadsheet file and the active document in WordPerfect. This command is the equivalent of choosing Spreadsheet from the Tools menu, then choosing Edit Link.

### **syntax**

SpreadsheetLinkEditDlg ( )

SpreadsheetLinkCreateDlg

SpreadsheetImportLink

---

## SpreadsheetLinkOptions

The SpreadsheetLinkOptions command specifies whether spreadsheet links will automatically be updated when a document containing a link is retrieved, and whether link codes will be displayed on the screen. This command is the equivalent of setting options in the Link Options [dialog box](#), then choosing OK.

### **syntax**

SpreadsheetLinkOptions (UpdateOnRetrieve:*Value Set Member*; ShowLinkCodes:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **UpdateOnRetrieve:**

This parameter specifies whether spreadsheet links will automatically be updated when a document containing a link is retrieved. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

#### **ShowLinkCodes:**

This parameter specifies whether link codes will be displayed on the screen. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

[SpreadsheetLinkOptionsDlg](#)



---

## SpreadsheetLinkOptionsDlg

The SpreadsheetLinkOptionsDlg command displays the Link Options dialog box, which lets the user specify whether spreadsheet links will automatically be updated when a document containing a link is retrieved, and whether link codes will be displayed on the screen. This command is the equivalent of choosing Spreadsheet from the Tools menu, then choosing Link Options.

### **syntax**

SpreadsheetLinkOptionsDlg ( )

SpreadsheetLinkOptions

---

## SpreadsheetLinkUpdateAll

The SpreadsheetLinkUpdateAll command updates any spreadsheet links in the current document. This command is the equivalent of choosing Spreadsheet from the Tools menu, then choosing Update All Links, then choosing Yes in the Update All Links dialog box.

### **syntax**

SpreadsheetLinkUpdateAll ( )

---

## SpreadsheetLinkUpdateAllDlg

The SpreadsheetLinkUpdateAllDlg command displays the Update Spreadsheet Links dialog box, which lets the user specify whether to update all of the spreadsheet links in the current document. This command is the equivalent of choosing Spreadsheet from the Tools menu, then choosing Update All Links.

### **syntax**

SpreadsheetLinkUpdateAllDlg ( )

SpreadsheetLinkUpdateAll

---

## Styles

Use the Styles command to turn a specified style on or off. This command is the equivalent of selecting a style in the Styles [dialog box](#), then choosing the On or Off [button](#).

### **syntax**

Styles (Name:*Character Expression*; State:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **Name:**

This character expression specifies the name of the style to be turned on or off. This character expression should correspond to a style that is currently available.

#### **State:**

This parameter specifies whether the style mentioned in the Name parameter will be turned on or off. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

[StylesDlg](#)

---

## StylesDelete

The StylesDelete command removes a specified style from the list of currently available styles, with the option of leaving formatting codes, deleting formatting codes, or deleting the definition only. This command is the equivalent of selecting a style in the Styles dialog box, choosing Delete, then choosing an option from the Delete Style dialog box.

### **syntax**

StylesDelete (Name:*Character Expression*; Type:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **Name:**

This character expression specifies the name of the style to be deleted. It should correspond to a style that is currently available.

#### **Type:**

This parameter specifies what happens to styles in the document when the specified style is deleted, corresponding to the options in the Delete Style dialog box.

The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Include!	0
----------	---

Leave!	1
--------	---

Definition!	2
-------------	---

---

## StylesDlg

The StylesDlg command displays the Styles dialog box, which lets the user create, edit, delete, turn on or off, retrieve, or save a style. This command is the equivalent of choosing Styles from the Layout menu.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

StylesDlg ( )

Styles

---

## StylesEdit

Use the StylesEdit command to create a style or edit an existing style. You can also use this command to create and edit outline styles.

### syntax

StylesEdit (Name:*Character Expression*; Description:*Character Expression*; StyleType:*Value Set Member*; EnterType:*Value Set Member*; Level:*Numeric Expression*; EditCodes:*Value Set Member*)

### parameters

This command has six parameters:

#### Name:

This character expression specifies the name of the style to be created or edited. If a style already exists by the name specified in this parameter, that style will be edited. If no style by that name exists, a style by that name will be created.

#### Description:

This character expression gives a brief description of the style named in the Name parameter. If the StylesEdit command is editing (rather than creating) a style, this character expression replaces any existing description of the style. If you do not want to overwrite the existing description, you may omit this parameter.

#### StyleType:

This parameter specifies whether the style is an open style or a paired style. The valid values are:

Member	Numeric Equivalent
--------	--------------------

Open!	0
Paired!	1

#### EnterType:

This parameter specifies the effect of pressing Enter when the style being created or edited is on. If the StylesEdit command is editing a style, this parameter replaces the existing Enter setting. If you do not want to replace this setting, omit this parameter. The valid values are:

Member	Numeric Equivalent
--------	--------------------

HardReturn!	0
Off!	1
OffOn!	2

#### Level:

This numeric expression specifies the outline level for the style being created or edited. If you do not want to create or edit an outline style, omit this parameter. This numeric expression should have a value from 1 to 8, corresponding to the number of outline levels available in WordPerfect.

#### EditCodes:

This parameter specifies whether to open the style editor. If this parameter is set to Yes!, the Style Editor is opened, and subsequent macro commands apply to the Style Editor until

it is closed. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

Styles

StylesProperties



---

## StylesProperties

The StylesProperties command sets the name, description, style type, and function of the Enter key for the style currently being edited in the Style Editor. This command should not be used outside of the Style Editor, especially not in the Outline Style Editor.

### **syntax**

StylesProperties (Name:*Character Expression*; Description:*Character Expression*;  
StyleType:*Value Set Member*; EnterType:*Value Set Member*)

### **parameters**

This command has four parameters:

#### **Name:**

This character expression overwrites the current style name. If you do not want to change the current name of the style being edited, you should omit this parameter.

#### **Description:**

This character expression replaces any existing description of the style. If you do not want to overwrite the existing description, you may omit this parameter.

#### **StyleType:**

This parameter specifies whether the style is an open style or a paired style. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Open!	0
-------	---

Paired!	1
---------	---

#### **EnterType:**

This parameter replaces the existing Enter setting. If you do not want to change the Enter setting, omit this parameter. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

HardReturn!	0
-------------	---

Off!	1
------	---

OffOn!	2
--------	---

[Styles](#)

[StylesEdit](#)

[StylesPropertiesDlg](#)

---

## StylesPropertiesDlg

The StylesPropertiesDlg command displays the Style Properties dialog box, which lets the user set the name, description, style type, and function of the Enter key for the style currently being edited in the Style Editor. This command should not be used outside of the Style Editor, especially not in the Outline Style Editor.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

StylesPropertiesDlg ( )

Styles

StylesEdit

StylesProperties

---

## StylesRetrieve

The StylesRetrieve command retrieves a specified list of files and makes those styles available to WordPerfect. This command is the equivalent of setting a full pathname in the Retrieve Styles dialog box.

### **syntax**

StylesRetrieve (Filename:*Character Expression*)

### **parameters**

This command has one parameter:

#### **Filename:**

This character expression specifies the full path of the list of styles to be retrieved.

StylesSave

---

## StylesSave

The StylesSave command saves the current list of styles in the file specified in the Filename parameter. This command is the equivalent of typing a name in the Save As text box in the Save Styles dialog box, then choosing Save.

### **syntax**

StylesSave (Filename:*Character Expression*)

### **parameters**

This command has one parameter:

#### **Filename:**

This character expression specifies the full path of the list of styles to be saved.

StylesRetrieve

---

## **Tab**

The Tab command inserts a tab at the insertion point in the active document. If the insertion point is in a table, the Tab command moves the insertion point to the next cell in the table. If you want to insert a tab in a table, use the TabInsert command. The Tab command is the equivalent of pressing Tab.

### **syntax**

Tab ( )

### TabInsert

---

## TabInsert

The TabInsert command inserts a tab at the insertion point in the active document. If the insertion point is in a table, the TabInsert command inserts a tab into the cell where the insertion point is located, rather than moving the insertion point to the next cell. The TabInsert command is the equivalent of pressing Ctrl+Tab.

### **syntax**

TabInsert ( )

---

## **TableAppendRow**

The TableAppendRow command inserts a new row below the current row. The current row is the row in the table containing the insertion point. This command is the equivalent of pressing Alt+Shift+Insert when the insertion point is in a table.

### **syntax**

TableAppendRow ( )

TableInsertRow

---

## TableBoxCaptionEdit

The TableBoxCaptionEdit command opens the Caption Editor for a specified table box, allowing subsequent commands to edit the caption text for that table box. If a table box is selected when this command is encountered, the Caption Editor opens for that box. If no table box is selected when the TableBoxCaptionEdit command is encountered, the macro pauses and the Edit Table Box Caption dialog box is displayed, which lets the user specify which table box's caption to edit. Once the user specifies a table box, the Table Caption Editor opens, and the macro resumes. You can close the Caption Editor with the Close command.

This command is not recordable. To use the TableBoxCaptionEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing Table Box from the Graphics menu, then choosing Caption, the macro facility records the BoxEditCaption ( ) command, which performs the same function.

### **syntax**

TableBoxCaptionEdit ( )

Close

BoxEditCaption



---

## TableBoxCreate

The TableBoxCreate command creates a table box and opens the Text Box Editor. The Text Box Editor can be closed with the Close command.

This command is not recordable. To use the TableBoxCreate command, you must add it to the macro by editing the macro.

### **syntax**

TableBoxCreate ( )

Close

TableBoxEdit

---

## TableBoxCreateDlg

The TableBoxCreateDlg command displays the Select Editor dialog box, which lets the user specify whether to create a table box in the Figure, Text, or Equation Editor. The TableBoxCreateDlg command is the equivalent of choosing Table Box from the Graphics menu, then choosing Create.

### **syntax**

TableBoxCreateDlg ( )

TableBoxEdit

---

## TableBoxEdit

The TableBoxEdit command opens the appropriate editor. If a table box is selected when this command is encountered, the appropriate editor opens with that box. If no table box is selected when the TableBoxEdit command is encountered, the macro pauses and the Edit Table Box dialog box is displayed, which lets the user specify which table box to edit. Once the user specifies a table box to edit, that box is brought into the appropriate editor, and the macro resumes.

This command is not recordable. To use the TableBoxEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing Table Box from the Graphics menu, then choosing Edit, the macro facility records the BoxEditor (Type:TableBox!) command, which performs the same function.

### **syntax**

TableBoxEdit ( )

Close

---

## **TableBoxNewNumberDlg**

The TableBoxNewNumberDlg command displays the Table Box Number dialog box, which lets the user set a new number from which table box numbers can increment. This command is the equivalent of choosing Table Box from the Graphics menu, then choosing New Number.

### **syntax**

TableBoxNewNumberDlg ( )

BoxNewNumber

---

## TableBoxOptionsDlg

The TableBoxOptionsDlg command displays the Table Box Options dialog box, which lets the user set border, spacing, caption, and other options for table boxes. This command is the equivalent of choosing Table Box from the Graphics menu, then choosing Options.

### **syntax**

TableBoxOptionsDlg ( )

BoxOptions

---

## TableBoxPositionDlg

The TableBoxPositionDlg command displays the Box Position and Size dialog box, which lets the user set the position, size, anchor, and other options for a specified table box. This command can open the Position and Size dialog box in several ways:

If a table box is being edited, the TableBoxPositionDlg command is the equivalent of choosing Box Position from the File menu in the Figure, Text, or Equation Editor.

If a table box is currently selected when the TableBoxPositionDlg command is encountered, the Position and Size dialog box is displayed, which lets the user set position and other options for that box.

If no box is selected, the Box Position and Size dialog box first appears, which lets the user specify a table box to modify. When the user has specified a number, the Position and Size dialog box is displayed, which lets the user set position and other options for that box.

### **syntax**

TableBoxPositionDlg ( )

### BoxPosition

---

## TableCalculate

The TableCalculate command calculates any formulas that have been defined in the current table. (The current table is the table containing the insertion point.) If the insertion point is not in a table, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. The TableCalculate command is the equivalent of choosing Tables from the Layout menu, then choosing Calculate.

### **syntax**

TableCalculate ( )

LABEL

ONERROR

TableFormula

TableFormulaDlg

---

## TableCell

The TableCell command sets attributes for the cell containing the insertion point, or, if cells are selected, for all selected cells. This command is the equivalent of setting options in the Format Cell dialog box.

### syntax

TableCell (Bold:*Value Set Member*; Underline:*Value Set Member*; DbUnderline:*Value Set Member*; Italic:*Value Set Member*; Outline:*Value Set Member*; Shadow:*Value Set Member*; SmallCap:*Value Set Member*; RedLine:*Value Set Member*; Strikeout:*Value Set Member*; Superscript:*Value Set Member*; Subscript:*Value Set Member*; Fine:*Value Set Member*; Small:*Value Set Member*; Large:*Value Set Member*; VeryLarge:*Value Set Member*; ExtraLarge:*Value Set Member*; ColumnAppSize:*Value Set Member*; Justify:*Value Set Member*; ColumnJustify:*Value Set Member*; NoCalc:*Value Set Member*; VertAlign:*Value Set Member*; Lock:*Value Set Member*; Shade:*Value Set Member*)

### parameters

This command has 23 parameters:

**Bold: Underline: DbUnderline: Italic: Outline: Shadow: SmallCap: Redline: StrikeOut: Superscript: Subscript: Fine: Small: Large: VeryLarge: ExtraLarge:**

Each of these parameters turns on or off the specified attribute for the current cell (or selected cells), corresponding to the options in the Format Cell dialog box. The members of each value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Off!	0
On!	1

### ColumnAppSize:

This parameter specifies whether the column size and appearance settings in the Format Column dialog box or the TableColumn command apply to the current cell (or selected cells). This parameter is the equivalent of either selecting or clearing the Use Column Size and Appearance check box in the Format Cell dialog box. The members and corresponding numeric equivalents of this value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Off!	0
On!	1

### Justify:

This parameter sets the justification for the current cell (or selected cells), corresponding to the options in the Justification pop-up list in the Format Cell dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Left!	0
Full!	1
Center!	2
Right!	3



DecimalAlign! 4

**ColumnJustify:**

This parameter specifies whether the justification setting specified in the Format Column dialog box or the TableColumn command applies to the current cell (or selected cells). This parameter is the equivalent of either selecting or clearing the Use Column Justification check box in the Format Cell dialog box. The valid values are:

Member	Numeric Equivalent
Off!	0
On!	1

**NoCalc:**

This parameter specifies whether the current cell (or selected cells) should be ignored when formulas in the current table are calculated. This parameter is the equivalent of either selecting or clearing the Ignore Cell When Calculating check box in the Format Cell dialog box. The valid values are:

Member	Numeric Equivalent	Definition
Off!	0	The current cell is not ignored when formulas are calculated.
On!	1	The current cell is ignored when formulas are calculated.

**VertAlign:**

This parameter specifies whether the vertical orientation of text for the current cell (or selected cell) should be along the top, bottom, or center of the cell. This parameter is the equivalent of choosing an option from the Alignment pop-up list in the Format Cell dialog box. The valid values are:

Member	Numeric Equivalent
Top!	0
Bottom!	1
Center!	2

**Lock:**

This parameter specifies whether information in the current cell (or selected cells) can be modified. When Lock is on, you cannot change the information in the cell. When lock is off, text and attributes can be changed. This parameter is the equivalent of either selecting or clearing the Lock check box in the Format Cell dialog box. The valid values are:

Member	Numeric Equivalent	Description
Off!	0	Text and attributes in table can be changed.
On!	1	Text and attributes in table cannot be changed.

**Shade:**

This parameter specifies whether the current cell (or selected cells) should be shaded. This parameter is the equivalent of either selecting or clearing the Shading check box in the Format Cell dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Off!	0
------	---

On!	1
-----	---

TableCellDlg

TableColumn

TableColumnDlg

---

## TableCellDlg

The TableCellDlg command displays the Format Cell dialog box, which lets the user set appearance, size, and other attributes for the current cell or selected cells in a table. The current cell is the cell containing the insertion point. This command is the equivalent of choosing Tables from the Layout menu, then choosing Cell.

### **syntax**

TableCellDlg ( )

TableCell

TableColumn

TableColumnDlg

---

## TableColumn

The TableColumn command sets appearance, size, justification, and other attributes for the current column (or selected columns). The current column is the table column containing the insertion point. This command is the equivalent of setting options in the Format Column dialog box.

### syntax

TableColumn (Bold:*Value Set Member*; Underline:*Value Set Member*; DbUnderline:*Value Set Member*; Italic:*Value Set Member*; Outline:*Value Set Member*; Shadow:*Value Set Member*; SmallCap:*Value Set Member*; RedLine:*Value Set Member*; Strikeout:*Value Set Member*; Superscript:*Value Set Member*; Subscript:*Value Set Member*; Fine:*Value Set Member*; Small:*Value Set Member*; Large:*Value Set Member*; VeryLarge:*Value Set Member*; ExtraLarge:*Value Set Member*; Digits:*Numeric Expression*; Justify:*Value Set Member*; Width:*Measurement Expression*)

### parameters

This command has 19 parameters:

#### **Bold: Underline: DbUnderline: Italic: Outline: Shadow: SmallCap: Redline: StrikeOut:**

Each of these parameters turns on or off the specified attribute for the current column (or selected columns), corresponding to the options in the Appearance group box in the Format Column dialog box. The members of each value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Off!	0
On!	1

#### **Superscript: Subscript:**

These parameters turn the specified attribute on or off for the current column (or selected columns), corresponding to the Superscript and Subscript checkboxes in the Format Column dialog box. If one of these parameters is set to On!, the other should not be set to On!, since superscript and subscript cannot both be on at the same time. The members of each value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Off!	0
On!	1

#### **Fine: Small: Large: VeryLarge: ExtraLarge:**

These parameters turn the specified attribute on or off for the current column (or selected columns), corresponding to the checkboxes in the Size group box in the Format Column dialog box. If one of these parameters is set to On!, the others should not be set to On!, since only one size attribute should be applied at a time. The members of each value set are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Off!	0
On!	1

**Digits:**

This numeric expression specifies the number of digits to be allowed to the right of a decimal point. This parameter is the equivalent of typing a number in the Digits box in the Format Column dialog box. The Digits parameter should be a positive integer.

**Justify:**

This parameter sets the justification for the current column (or selected columns), corresponding to the options in the Justification pop-up list in the Format Column dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Left!	0
Full!	1
Center!	2
Right!	3
DecimalAlign!	4

**Width:**

This measurement expression sets the column width for the current column (or selected columns). This parameter is the equivalent of setting a measurement in the Column Width box in the Format Column dialog box.

TableCell

TableCellDlg

TableColumnDlg

---

## TableColumnDlg

The TableColumnDlg command displays the Format Column dialog box, which lets the user set appearance, size, justification, and other attributes for the current column (or selected columns). The current column is the table column containing the insertion point. This command is the equivalent of choosing Tables from the Layout menu, then choosing Column.

### **syntax**

TableColumnDlg ( )

TableCell

TableCellDlg

TableColumn

---

## TableConvert

The TableConvert command converts selected columns (parallel or tabular) of text into a table. This command is ignored if no columns are selected. The TableConvert command is the equivalent of choosing Tables from the Layout Menu while columns of text are selected, then choosing Create, then choosing Tabular Column or Parallel Column from the Convert Table dialog box, then choosing OK.

### **syntax**

TableConvert (ConvertFrom:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **ConvertFrom:**

This parameter specifies whether the selection should be converted from tabular or parallel columns. This parameter is the equivalent of choosing one of the options in the Convert Table dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

TabularColumns!	0
ParallelColumns!	1

TableConvertDlg

---

## TableConvertDlg

The TableConvertDlg command displays the Convert Table dialog box, which lets the user convert selected tabular and parallel columns into a table. If no text is selected, this command terminates the macro or redirects macro execution to the LABEL command specified by an ONERROR command. If text is selected, the TableConvertDlg command is the equivalent of choosing Tables from the Layout menu, then choosing Create.

### **syntax**

TableConvertDlg ( )

LABEL

ONERROR

TableConvert



---

## TableCreate

The TableCreate command creates a table with a specified number of rows and columns at the insertion point in the active document. This command is the equivalent of setting options in the Create Table dialog box.

### **syntax**

TableCreate (Columns:*Numeric Expression*; Rows:*Numeric Expression*)

### **parameters**

This command has two parameters:

#### **Columns: Rows:**

These numeric expressions specify how many rows and columns should be in the table. These numeric expressions should be positive integers.

TableCreateNewDlg

---

## TableCreateNewDlg

The TableCreateNewDlg command displays the Create Table dialog box, which lets the user create a table with a specified number of rows and columns. This command is the equivalent of choosing Tables from the Layout menu, then choosing Create.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

TableCreateNewDlg ( )

TableCreate

---

## TableDeleteColumn

The TableDeleteColumn command deletes the current table column, selected table columns, or a specified number of table columns. The TableDeleteColumn command is the equivalent of specifying a number of columns to delete in the Columns text box of the Delete Columns/Rows dialog box.

### **syntax**

TableDeleteColumn (Count:*Numeric Expression*)

### **parameters**

This command has one parameter:

#### **Count:**

This numeric expression specifies how many columns to delete in the table. WordPerfect deletes columns beginning with the current column and moving right. If this parameter is omitted, the TableDeleteColumn command deletes the current column (the column containing the insertion point) or the selected columns in the table. This numeric expression should be a positive integer.

TableDeleteDlg

TableDeleteRow

---

## **TableDeleteDlg**

The TableDeleteDlg command displays the Delete Rows/Columns dialog box, which lets the user specify how many rows or columns to remove from the current table. The TableDeleteDlg command is the equivalent of choosing Tables from the Layout menu, then choosing Delete.

### **syntax**

TableDeleteDlg ( )

TableDeleteColumn

TableDeleteRow

---

## TableDeleteRow

The TableDeleteRow command deletes the current table row, selected table rows, or a specified number of table rows. The TableDeleteRow command is the equivalent of specifying a number of rows to delete in the Rows text box of the Delete Columns/Rows dialog box.

### **syntax**

TableDeleteRow (Count:*Numeric Expression*)

### **parameters**

This command has one parameter:

#### **Count:**

This numeric expression specifies the number of rows to delete in the table. WordPerfect deletes rows beginning with the current row and moving down. If this parameter is omitted, the TableDeleteRow command deletes the current row (the row containing the insertion point) or the selected rows in the table. This numeric expression should be a positive integer.

TableDeleteColumn

TableDeleteDlg

---

## TableEditColumn

The TableEditColumn command sets the left edge and column widths for the current table. (The current table is the table containing the insertion point.) This command is the equivalent of setting the left margin and column stops on the ruler while the insertion point is in a table.

### **syntax**

TableEditColumn (LeftEdge:*Measurement Expression*; Width1:*Measurement Expression*;  
Width2:*Measurement Expression*; Width3:*Measurement Expression*; Width4:*Measurement Expression*;  
Width5:*Measurement Expression*; Width6:*Measurement Expression*;  
Width7:*Measurement Expression*; Width8:*Measurement Expression*; Width9:*Measurement Expression*;  
Width10:*Measurement Expression* ; Width11:*Measurement Expression*;  
Width12:*Measurement Expression*; Width13:*Measurement Expression*;  
Width14:*Measurement Expression*; Width15:*Measurement Expression*;  
Width16:*Measurement Expression*; Width17:*Measurement Expression*;  
Width18:*Measurement Expression*; Width19:*Measurement Expression*;  
Width20:*Measurement Expression*; Width21:*Measurement Expression*;  
Width22:*Measurement Expression*; Width23:*Measurement Expression*;  
Width24:*Measurement Expression*; Width25:*Measurement Expression*;  
Width26:*Measurement Expression*; Width27:*Measurement Expression*;  
Width28:*Measurement Expression*; Width29:*Measurement Expression*;  
Width30:*Measurement Expression*; Width31:*Measurement Expression*;  
Width32:*Measurement Expression*)

### **parameters**

This command has 33 parameters. All of the parameters are optional. For example, if you want to move the table left or right, then use only the LeftEdge parameter. If you want to change the width of column 2, use only the Width2 parameter.

#### **LeftEdge:**

This measurement expression sets the distance from the left margin to the left edge of the table.

**Width1: Width2: Width3: Width4: Width5: Width6: Width7: Width8: Width9:  
Width10: Width11: Width12: Width13: Width14: Width15: Width16: Width17:  
Width18: Width19: Width20: Width21: Width22: Width23: Width24: Width25:  
Width26: Width27: Width28: Width29: Width30: Width31: Width32:**

These measurement expressions set the width of their respective table columns. They do not set the distance of the column stop from the left margin or left edge of the table. You only need to include Width parameters for the columns the current table contains.

---

## TableFormula

The TableFormula command inserts a formula into the current cell, or specified cells, of the current table. This command is the equivalent of setting options in the Tables Formula dialog box.

### **syntax**

TableFormula (Formula:*Character Expression*; Copy:*Value Set Member*; Cell:*Character Expression*; Count:*Numeric Expression*)

### **parameters**

This command has four parameters:

#### **Formula:**

This character expression specifies the formula to be calculated.

#### **Copy:**

This command specifies whether the formula specified in the Formula parameter will be copied to a particular cell to several cells in the current row or column. This parameter is the equivalent of choosing one of the radio buttons in the Tables Formula dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Description</b>
To!	0	Copy a formula to a cell specified in the Cell parameter.
Down!	1	Copy a formula down from the current cell to the number of cells specified in the Count parameter.
Right!	2	Copy a formula right from the current cell to the number of cells specified in the Count parameter.

#### **Cell:**

This character expression specifies the cell to which the formula should be copied. This parameter should only be used if the Copy parameter is set to To!.

#### **Count:**

This numeric expression specifies how many occurrences of the formula should be copied into the table. The copies of the formula are copied down from the current cell, or to the right of the current cell, depending on the value of the Copy parameter. If the Copy parameter is set to To!, the Count parameter should be omitted.

TableFormulaDlg

---

## TableFormulaDlg

The TableFormulaDlg command displays the Tables Formula dialog box, which lets the user insert a formula into the current cell or into specified cells of the current table. This command is the equivalent of choosing Tables from the Layout menu, then choosing Formula.

### **syntax**

TableFormulaDlg ( )

TableFormula



---

## TableInsertColumn

The TableInsertColumn command inserts a specified number of new columns to the right of the current column. The current column is the table column containing the insertion point. This command is the equivalent of choosing Columns and specifying a number in the Insert Rows/Columns dialog box.

### **syntax**

TableInsertColumn (Count:*Numeric Expression*)

### **parameters**

This command has one parameter. If this parameter is omitted, the TableInsertColumn command inserts a single column.

### **Count:**

This numeric expression specifies how many columns to insert to the right of the current column. This numeric expression should be a positive integer.

TableInsertDlg

TableInsertRow

---

## TableInsertDlg

The TableInsertDlg command displays the Insert Rows/Columns dialog box, which lets the user specify how many rows or columns to insert into the current table. The TableInsertDlg command is the equivalent of choosing Tables from the Layout menu, then choosing Insert.

### **syntax**

TableInsertDlg ( )

TableInsertColumn

TableInsertRow

---

## TableInsertRow

The TableInsertRow command inserts a specified number of new rows above the current row. (The current row is the row in the table containing the insertion point.) This command is the equivalent of choosing Rows and specifying a number in the Insert Rows/Columns dialog box.

### **syntax**

TableInsertRow (Count:*Numeric Expression*)

### **parameters**

This command has one parameter. If this parameter is omitted, the TableInsertRow command inserts a single row.

#### **Count:**

This numeric expression specifies how many rows to insert above the current row. This numeric expression should be a positive integer.

TableAppendRow

TableInsertColumn

TableInsertDlg

---

## TableJoin

The TableJoin command joins selected cells in a table into a single cell. If no cells are selected, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. The TableJoin command is the equivalent of choosing Tables from the Layout menu, then choosing Join.

### **syntax**

TableJoin ( )

LABEL

ONERROR

---

## TableLineDlg

The TableLineDlg command displays the Table Lines dialog box, which lets the user specify the appearance of the current cell or selected cells in a table. This command is the equivalent of choosing Tables from the Layout menu, then choosing Lines.

### **syntax**

TableLineDlg ( )

TableLines

---

## TableLines

The TableLines command specifies the appearance of the current cell or selected cells in a table. This command is the equivalent of setting options in the Table Lines [dialog box](#).

### **syntax**

TableLines (Left:*Value Set Member*; Right:*Value Set Member*; Top:*Value Set Member*; Bottom:*Value Set Member*; Inside:*Value Set Member*; Outside:*Value Set Member*)

### **parameters**

This command has six parameters:

#### **Left: Right: Top: Bottom: Inside: Outside:**

These parameters specify how the lines will appear on each side of a cell, corresponding to the pop-up lists in the Table Lines dialog box. The members of these value sets are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

None!	0
Single!	1
Double!	2
Dashed!	3
Dotted!	4
Thick!	5
ExtraThick!	6

[TableLineDlg](#)

---

## TableOptions

The TableOptions command sets the size, margins, position, headers, shading attributes, negative result display, and cell lock capability for the current table. (The current table is the table containing the [insertion point](#).) This command is the equivalent of setting options in the Table Options [dialog box](#).

### syntax

TableOptions (Columns:*Numeric Expression*; Rows:*Numeric Expression*;  
LeftMargin:*Measurement Expression*; RightMargin:*Measurement Expression*;  
TopMargin:*Measurement Expression*; BottomMargin:*Measurement Expression*;  
TablePosition:*Value Set Member*; FromLeftEdge:*Measurement Expression*;  
HeaderRows:*Numeric Expression*; ShadePercent:*Numeric Expression*; NegDisplay:*Value Set Member*; DisableLocks:*Value Set Member*)

### parameters

This command has 12 parameters:

#### Columns: Rows:

These numeric expressions specify how many columns and rows should be in the current table. If the number of columns or rows specified in this parameter is different than the number of columns and rows currently in the table, WordPerfect will add or remove the necessary columns and rows. Columns are added to or removed from the right side of the table. Rows are added to or removed from the bottom of the table.

The Columns and Rows parameters are the equivalent of typing numbers in the Rows and Columns boxes in the Table Options dialog box. These numeric expressions should be positive integers.

#### LeftMargin: RightMargin: TopMargin: BottomMargin:

These measurement expressions set the amount of inside border space for each cell in the table. These parameters are the equivalent of typing measurements in the Left, Right, Top, and Bottom boxes in the Table Options dialog box.

#### TablePosition:

This parameter specifies the horizontal location of the table on the page, corresponding to the options in the Position [group box](#) in the Table Options dialog box. The valid values are:

Member	Numeric Equivalent
--------	--------------------

Left!	0
Right!	1
Center!	2
Full!	3
FromLeftEdge!	4

#### FromLeftEdge:

Use this measurement expression if you specify FromLeftEdge! as the value set member for the TablePosition parameter. The FromLeftEdge parameter specifies the distance from the left margin of the page to the left edge of the table.

#### HeaderRows:

This numeric expression specifies how many rows (starting at the top row and working down) of the table should be used as "header rows". Header rows appear at the top of each

page of tables that span more than one page. You may omit the HeaderRows parameter or set it to 0 if the table does not need a header.

**ShadePercent:**

This numeric expression specifies how dark to shade any cells in the current table. Note that this parameter affects only cells that are already defined as shaded. Use the TableCell command to shade parts of a table.

**NegDisplay:**

This parameter specifies whether negative numbers (results of calculations) will be enclosed within parentheses or preceded by a minus sign (-). This parameter is the equivalent of choosing Minus Sign or Parentheses in the Table Options dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
Minus!	0
Paren!	1

**DisableLocks:**

This parameter specifies whether it will be possible to lock cells in the current table. When a cell is locked, you cannot change text and attributes in the cell. This parameter is the equivalent of either selecting or clearing the Disable Cell Locks check box in the Table Options dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
Off!	0	Cell locks are not disabled.
On!	1	Cell locks are disabled.

TableCell

TableCellDlg

TableOptionsDlg



---

## TableOptionsDlg

The TableOptionsDlg command displays the Table Options dialog box, which lets the user set the size, margins, position, headers, shading attributes, negative result display, and cell lock capability for the current table. This command is the equivalent of putting the insertion point in a table, then choosing Tables from the Layout menu, then choosing Options. If the insertion point is not in a table when this command is encountered, macro execution is terminated or redirected to the LABEL specified by an ONERROR command.

### **syntax**

TableOptionsDlg ( )

LABEL

ONERROR

TableCell

TableCellDlg

TableOptions

---

## TableRow

The TableRow command specifies the height for the current row or selected rows. The TableRow command also specifies whether more than one line of text will be allowed in the current row or selected rows. This command is the equivalent of setting options in the Format Row [dialog box](#).

### **syntax**

TableRow (SingleLine:*Value Set Member*; FixedHeight:*Value Set Member*; Height:*Measurement Expression*)

### **parameters**

This command has three parameters:

#### **SingleLine:**

This parameter specifies whether a single line of text or multiple lines of text will be allowed in the current row or selected rows. This parameter is the equivalent of choosing Single Line or Multi Line in the Format Row dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

Off!	0
On!	1

#### **FixedHeight:**

This parameter specifies whether the height of the row should be set to a fixed height, or if WordPerfect should automatically adjust row height according to the size of the text in the row. This parameter is the equivalent of choosing Fixed or Auto in the Format Row dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Description</b>
---------------	---------------------------	--------------------

Off!	0	WordPerfect adjusts row height according to the size of the text in the row.
On!	1	WordPerfect sets row height to the setting specified in the Height parameter.

#### **Height:**

This measurement expression specifies the height of the row (or selected rows) if the FixedHeight parameter is set to On!. If the FixedHeight parameter is set to Off!, the Height parameter should be omitted.

[TableCell](#)

[TableRowDlg](#)

---

## TableRowDlg

The TableRowDlg command displays the Format Row dialog box, which lets the user specify the row height for a table and whether multiple lines are allowable in the current row or selected rows. (The current row is the table row containing the insertion point.) If a table does not already exist, or if the insertion point is not in a table, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command. This command is the equivalent of choosing Tables from the Layout menu, then choosing Row.

### **syntax**

TableRowDlg ( )

LABEL

ONERROR

TableCellDlg

TableColumnDlg

---

## TableSplit

The TableSplit command splits the current cell or selected cells into a specified number of columns or rows. This command is the equivalent of setting options in the Split Column/Row [dialog box](#).

### **syntax**

TableSplit (Split:*Value Set Member*; Count:*Numeric Expression*)

### **parameters**

This command has two parameters:

#### **Split:**

This parameter specifies whether the current cell (or selected cells) will be split into columns or rows. This parameter is the equivalent of choosing Column or Row in the Split Column/Row dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Rows!	0
-------	---

Columns!	1
----------	---

#### **Count:**

This numeric expression specifies the number of cells into which the current cell or each selected cell should be divided. This parameter should be a positive integer.

[TableSplitDlg](#)

---

## TableSplitDlg

The TableSplitDlg command displays the Split Column/Row dialog box, which lets the user split the current cell or selected cells in a table into more columns or rows. This command is the equivalent of choosing Tables from the Layout command, then choosing Split.

### **syntax**

TableSplitDlg ( )

TableSplit

---

## TabSet

Use the TabSet command to create a tab setting that is effective beginning at the insertion point and that replaces any existing tab settings. You can specify as many as 40 tab stops, each with a specified position and tab type. The TabSet command is the equivalent of setting individual tabs in the Tab Set dialog box.

### syntax

TabSet (Origin:Value Set Member; Type1:Value Set Member; Pos1:Measurement Expression; Type2:Value Set Member; Pos2:Measurement Expression; Type3:Value Set Member; Pos3:Measurement Expression; Type4:Value Set Member; Pos4:Measurement Expression; Type5:Value Set Member; Pos5:Measurement Expression; Type6:Value Set Member; Pos6:Measurement Expression; Type7:Value Set Member; Pos7:Measurement Expression; Type8:Value Set Member; Pos8:Measurement Expression; Type9:Value Set Member; Pos9:Measurement Expression; Type10:Value Set Member; Pos10:Measurement Expression; Type11:Value Set Member; Pos11:Measurement Expression; Type12:Value Set Member; Pos12:Measurement Expression; Type13:Value Set Member; Pos13:Measurement Expression; Type14:Value Set Member; Pos14:Measurement Expression; Type15:Value Set Member; Pos15:Measurement Expression; Type16:Value Set Member; Pos16:Measurement Expression; Type17:Value Set Member; Pos17:Measurement Expression; Type18:Value Set Member; Pos18:Measurement Expression; Type19:Value Set Member; Pos19:Measurement Expression; Type20:Value Set Member; Pos20:Measurement Expression; Type21:Value Set Member; Pos21:Measurement Expression; Type22:Value Set Member; Pos22:Measurement Expression; Type23:Value Set Member; Pos23:Measurement Expression; Type24:Value Set Member; Pos24:Measurement Expression; Type25:Value Set Member; Pos25:Measurement Expression; Type26:Value Set Member; Pos26:Measurement Expression; Type27:Value Set Member; Pos27:Measurement Expression; Type28:Value Set Member; Pos28:Measurement Expression; Type29:Value Set Member; Pos29:Measurement Expression; Type30:Value Set Member; Pos30:Measurement Expression; Type31:Value Set Member; Pos31:Measurement Expression; Type32:Value Set Member; Pos32:Measurement Expression; Type33:Value Set Member; Pos33:Measurement Expression; Type34:Value Set Member; Pos34:Measurement Expression; Type35:Value Set Member; Pos35:Measurement Expression; Type36:Value Set Member; Pos36:Measurement Expression; Type37:Value Set Member; Pos37:Measurement Expression; Type38:Value Set Member; Pos38:Measurement Expression; Type39:Value Set Member; Pos39:Measurement Expression; Type40:Value Set Member; Pos40:Measurement Expression)

### parameters

This command has 81 parameters. You should specify a parameter for each tab stop you set. A Typex (where x is a number from 1 to 40) should precede each Posx parameter.

#### Origin:

This parameter specifies whether tab settings will be measured from the left edge of the page or from the left margin. The valid values are:

Member	Numeric Equivalent
--------	--------------------

---

LeftEdge!	0
LeftMargin!	1

**Type1: Type2: Type3: Type4: Type5: Type6: Type7: Type8: Type9: Type10: Type11:  
Type12: Type13: Type14: Type15: Type16: Type17: Type18: Type19: Type20:  
Type21: Type22: Type23: Type24: Type25: Type26: Type27: Type28: Type29:  
Type30: Type31: Type32: Type33: Type34: Type35: Type36: Type37: Type38:**

**Type39: Type40:**

Each Typex parameter specifies how text should be aligned on the tab stop being created. These parameters correspond to the options in the Tabs group box in the Tab Set dialog box. The members of each value set are:

**Member      Numeric Equivalent**

---

Left!	0
Center!	1
Right!	2
Decimal!	3
LeftDot!	4
CenterDot!	5
RightDot!	6
DecimalDot!	7

**Pos1: Pos2: Pos3: Pos4: Pos5: Pos6: Pos7: Pos8: Pos9: Pos10: Pos11: Pos12:  
Pos13: Pos14: Pos15: Pos16: Pos17: Pos18: Pos19: Pos20: Pos21: Pos22: Pos23:  
Pos24: Pos25: Pos26: Pos27: Pos28: Pos29: Pos30: Pos31: Pos32: Pos33: Pos34:  
Pos35: Pos36: Pos37: Pos38: Pos39: Pos40:**

Each Posx measurement expression specifies the location of the tab stop mentioned in the preceding Typex parameter.

TabSetDlg

---

## TabSetDlg

The TabSetDlg command displays the Tab Set dialog box, which lets the user specify the type and positions of tab stops. This command is the equivalent of choosing Line from the Layout menu, then choosing Tab Set.

### **syntax**

TabSetDlg ( )

TabSet



---

## TextBoxCaptionEdit

The TextBoxCaptionEdit command opens the Caption Editor for a selected text box allowing subsequent commands to edit the caption text for that text box.

If a text box is selected when this command is encountered, the Caption Editor opens for that box. If no text box is selected when the TextBoxCaptionEdit command is encountered, the macro pauses and the Edit Text Box Caption dialog box is displayed, which lets the user specify which text box caption to edit. Once the user specifies a text box, the Text Caption Editor opens, and the macro resumes. You can close the Caption Editor with the Close command.

This command is not recordable. To use the TextBoxCaptionEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing Text Box from the Graphics menu, then choosing Caption, the macro facility records the BoxEditCaption ( ) command, which performs the same function.

### **syntax**

TextBoxCaptionEdit ( )

Close

BoxEditCaption

---

## **TextBoxCreate**

The TextBoxCreate command creates a text box and opens the Text Box Editor. The Text Box Editor can be closed using the Close command. The TextBoxCreate command is the equivalent of choosing Text Box from the Graphics menu, then choosing Create.

### **syntax**

TextBoxCreate ( )

Close

TextBoxEdit

---

## TextBoxCreateDlg

The TextBoxCreateDlg command displays the Select Editor dialog box, which lets the user specify whether the text box being created will contain a figure, text, or an equation. When the user chooses OK, the appropriate editor opens, and the user can insert the contents of the box. The macro pauses until the user closes the editor, and then the macro resumes.

This command has no interface equivalent. This dialog box can only be invoked from a macro.

### **syntax**

TextBoxCreateDlg ( )

EquationCreateDlg

FigureCreateDlg

---

## **TextBoxEdit**

The TextBoxEdit command opens the Text Box Editor. If a text box is selected when this command is encountered, the Text Box Editor opens with that box. If no text box is selected when the TextBoxEdit command is encountered, the macro pauses and the Edit Text Box dialog box is displayed, which lets the user specify which text box to edit. Once the user specifies a text box to edit, that box is brought into the Text Box Editor, and the macro resumes.

This command is not recordable. To use the TextBoxEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing Text Box from the Graphics menu, then choosing Edit, the macro facility records the BoxEditor (Type:TextBox!) command, which performs the same function.

### **syntax**

TextBoxEdit ( )

Close

---

## **TextBoxNewNumberDlg**

The TextBoxNewNumberDlg command displays the Text Box Number dialog box, which lets the user set a new number from which text box numbers will increment. This command is the equivalent of choosing Text Box from the Graphics menu, then choosing New Number.

### **syntax**

TextBoxNewNumberDlg ( )

BoxNewNumber

---

## **TextBoxOptionsDlg**

The TextBoxOptionsDlg command displays the Text Box Options dialog box, which lets the user set border, spacing, caption, and other options for text boxes. This command is the equivalent of choosing Text Box from the Graphics menu, then choosing Options.

### **syntax**

TextBoxOptionsDlg ( )

BoxOptions

---

## TextBoxPositionDlg

The TextBoxPositionDlg command displays the Box Position and Size dialog box, which lets the user set the position, size, anchor, and other options for a specified text box. This command can open the Position and Size dialog box in several ways:

If a text box is being edited, the TextBoxPositionDlg command is the equivalent of choosing Box Position from the File menu in the Figure, Text, or Equation Editor.

If a text box is currently selected when the TextBoxPositionDlg command is encountered, the Position and Size dialog box is displayed, which lets the user set position and other options for that box.

If no box is selected, the Box Position and Size dialog box appears. This lets the user specify a number in the Text Box Number text box indicating which box to modify. When the user has specified a number, the Box Position and Size dialog box is displayed, which lets the user set position and size options for that box.

### **syntax**

TextBoxPositionDlg ( )

BoxSelect

---

## TextBoxRotate

The TextBoxRotate command rotates the text in the Text Box Editor a specified number of degrees. This command is the equivalent of choosing Rotate to open the Rotate Text [dialog box](#), then choosing one of the options in the Rotate Text Box dialog box. This command should be encountered when the Text Box Editor is [active](#).

### **syntax**

TextBoxRotate (RotationValue:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **RotationValue:**

This parameter specifies the degree to which the text in the Text Box Editor will be rotated. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Rotate0!	0
Rotate90!	1
Rotate180!	2
Rotate270!	3

[TextBoxCreate](#)

[TextBoxEdit](#)

[TextBoxRotateDlg](#)



---

## TextBoxRotateDlg

The TextBoxRotateDlg command displays the Rotate Text Box dialog box, which lets the user rotate the text in the Text Box Editor a specified number of degrees. This command is the equivalent of choosing Rotate in the Text Box Editor. This command should be encountered when the Text Box Editor is active.

### **syntax**

TextBoxRotateDlg ( )

TextBoxCreate

TextBoxEdit

TextBoxRotate

---

## Thesaurus

The Thesaurus command opens the Thesaurus program, which lets the user look up synonyms and antonyms for the word at the insertion point. The macro then pauses until the user chooses Close or Replace. The Thesaurus command is the equivalent of choosing Thesaurus from the Tools menu.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

Thesaurus ( )

Grammatik.

Speller

---

## ToADefine

The ToADefine command places a marker at the insertion point in the active document, which specifies where a section of the table of authorities will be generated. The ToADefine command is the equivalent of setting options in the Define Table of Authorities dialog box.

### **syntax**

ToADefine (SectionNumber:*Numeric Expression*; BlankLineBetween:*Value Set Member*; DotLeaders:*Value Set Member*; AllowUnderline:*Value Set Member*)

### **parameters**

This command has four parameters:

#### **SectionNumber:**

This numeric expression specifies which section of the table of authorities will be generated at the point of definition. This numeric expression should be an integer from 1 to 16.

#### **BlankLineBetween:**

This parameter specifies whether there will be an extra hard return between each authority in the table. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

#### **DotLeaders:**

This parameter specifies whether dot leaders should extend from the authority to its page number reference. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

#### **AllowUnderline:**

This parameter specifies whether underlining in the full form will be allowed in the table. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

ToAEditFull

ToAMarkFull

ToAMarkShort

ToADefineDlg

---

## ToADefineDlg

The ToADefineDlg command displays the Define Table of Authorities dialog box, which lets the user specify a section number and the numbering format for the table. When the user chooses OK, a Define Table of Authorities code is placed at the insertion point in the active document. The ToADefineDlg command is the equivalent of choosing Define from the Tools menu, then choosing Table of Authorities.

### **syntax**

ToADefineDlg ( )

ToADefine

---

## ToAEditFull

The ToAEditFull command opens a text editor containing the first table of authorities full form mark that precedes the insertion point. The text editor can be closed with the Close command.

If no marks precede the insertion point, this command opens the closest following full form mark into the text editor. If no full form marks exist in the active document, this command has no effect.

The ToAEditFull command is the equivalent of choosing Mark Text from the Tools menu, then choosing ToA Edit Full Form.

### **syntax**

ToAEditFull ( )

ToADefine

ToAMarkFull

ToAMarkShort

---

## ToAMarkFull

The ToAMarkFull command lets the user mark selected text for a table of authorities. If no text is selected, this command is ignored.

This command displays the Mark ToA Full Form dialog box, which lets the user specify the short form text and section number. When the user chooses OK, WordPerfect opens a text editor, which lets subsequent macro commands modify the full form text. Use the Close command to close the text editor.

This command is the equivalent of choosing Mark Text from the Tools menu, then choosing ToA Full Form.

### **syntax**

ToAMarkFull ( )

ToADefine

ToAEditFull

ToAMarkShort

---

## ToAMarkShort

The ToAMarkShort command sets a mark at the insertion point for a specified short form in a table of authorities. This command is the equivalent of specifying a short form in the Mark ToA Short Form dialog box.

### **syntax**

ToAMarkShort (ShortFormText:*Character Expression*)

### **parameters**

This command has one parameter:

#### **ShortFormText:**

This character expression specifies the name of the short form for the authority.

ToADefine

ToAEditFull

ToAMarkFull

ToAMarkShortDlg

---

## ToAMarkShortDlg

The ToAMarkShortDlg command displays the Mark ToA Short Form dialog box, which lets the user set the short form name for a table of authorities citation. This command is the equivalent of choosing Mark Text from the Tools menu, then choosing ToA Short Form.

### **syntax**

ToAMarkShortDlg ( )

ToAMarkShort



---

## ToCDefine

The ToCDefine command places a Define Table of Contents code at the insertion point in the active document. This command is the equivalent of setting options in the Define Table of Contents dialog box.

### **syntax**

ToCDefine (NumLevels:*Numeric Expression*; WrapLastLvl:*Value Set Member*;  
Level1Style:*Value Set Member*; Level2Style:*Value Set Member*; Level3Style:*Value Set Member*;  
Level4Style:*Value Set Member*; Level5Style:*Value Set Member*)

### **parameters**

This command has seven parameters:

#### **NumLevels:**

This numeric expression specifies how many levels should be available for the table of contents. This parameter is the equivalent of setting the Number of Levels box in the Define Table of Contents dialog box. The numeric expression should be a positive integer from 1 to 5.

#### **WrapLastLvl:**

This parameter specifies whether the final level of the table of contents will be wrapped. This parameter is the equivalent of choosing the Last Level in the Wrapped Format checkbox in the Define Table of Contents dialog box. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
Yes!	1

#### **Level1Style: Level2Style: Level3Style: Level4Style: Level5Style:**

These parameters specify the appearance of table of contents entries for each specified level. This is the equivalent of setting the pop-up lists in the Define Table of Contents dialog box. The members of these value sets are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

NoNumbering!	0
NumFollowsEntry!	1
FollowsInParens!	2
FlushRight!	3
FlushLeaders!	4

ToCDefineDlg

ToCMark

---

## ToCDefineDlg

The ToCDefineDlg command displays the Define Table of Contents dialog box, which lets the user specify the number of levels and the numbering format for the table. When the user chooses OK, a Define Table of Contents code is placed at the insertion point in the active document. The ToCDefineDlg command is the equivalent of choosing Define from the Tools menu, then choosing Table of Contents.

### **syntax**

ToCDefineDlg ( )

ToCDefine

ToCMark

---

## ToCMark

The ToCMark command marks selected text for a specified level in a table of contents. If no text is selected, this command is ignored. The ToCMark command is the equivalent of specifying a level in the Mark Table of Contents [dialog box](#).

### **syntax**

ToCMark (LevelNumber:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **LevelNumber:**

This parameter specifies on which level the marked text will appear. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Level1!	0
Level2!	1
Level3!	2
Level4!	3
Level5!	4

[ToCDefine](#)

[ToCMarkDlg](#)

---

## ToCMarkDlg

The ToCMarkDlg command displays the Mark Table of Contents dialog box, which lets the user insert a table of contents mark. This command is the equivalent of choosing Mark Text from the Tools menu, then choosing Table of Contents.

### **syntax**

ToCMarkDlg ( )

ToCDefine

ToCMark

---

## Type

The Type command types a specified character expression into the active document, beginning at the insertion point. This command is the equivalent of typing text in a document window.

The Type command is also useful if you want to type the value of a variable into a WordPerfect document. For instance, if you want to type the value of a variable called "surname" into a document, the type command would appear as:

```
Type (surname)
```

Keep in mind that the Type command can only type out character expressions--not numeric expressions. If you want to type the value of a variable containing a numeric expression into a document, you must first convert the numeric expression into a character expression. Then you can use the Type command to type the character expression into the document. For instance, if the variable "cost" contains the numeric value 17.23, and you want to type that value into a document, the macro would need to contain this sequence of commands:

```
NUMSTR (CostChars; 2; cost)
```

```
//The numeric value of cost variable to two decimal places is converted to a character expression, which is assigned to the variable CostChars.
```

```
WP.Type (CostChars)
```

```
//This command assumes the product prefix is WP.
```

### **syntax**

```
Type (Text:Character Expression)
```

### **parameters**

This command has one parameter:

#### **Text:**

This character expression specifies the text to be typed at the insertion point or the variable that holds the text. All character strings must be housed in quotation marks; otherwise, the character expression is treated as a variable.

Character expressions should not exceed 512 bytes. Each standard character is one byte in length; extended characters require more bytes.

A quotation mark that needs to be included as part of a character string must be preceded by an additional quotation mark.

NUMSTR

TypeChar

---

## TypeChar

The TypeChar command inserts a specified character into the current window. You can specify the character using a single number (see the Character parameter below) or using a WordPerfect character set number and character number.

### **syntax**

TypeChar (Character:*Numeric Expression*; CharacterSet:*Numeric Expression*;  
CharacterOffset:*Numeric Expression*)

### **parameters**

This command has three parameters:

#### **Character:**

This numeric expression should be an integer. This single number represents the character to be typed through a formula:

$(\text{Character Set Number} * 256) + \text{Character Number}$

So, if you want TypeChar to insert a check mark , which is character number 23 in character set 5, the Character parameter would be 1303, since  $(5 * 256) + 23 = 1303$ .

If you use this parameter, you should omit the CharacterSet and CharacterOffset parameters.

#### **CharacterSet:**

This numeric expression should be an integer. The CharacterSet parameter specifies the character set to which the character to be inserted belongs. This parameter should be used in conjunction with the CharacterOffset parameter. If the CharacterSet and CharacterOffset parameters are used, the Character parameter should be omitted.

#### **CharacterOffset:**

This numeric expression should be an integer. The CharacterOffset parameter specifies the Character number in the specified Character Set to be inserted. This parameter should be used in conjunction with the CharacterSet parameter. If the CharacterSet and CharacterOffset parameters are used, the Character parameter should be omitted.

Type

---

## TypesetBaseline

The TypesetBaseline command specifies whether the first baseline is placed even with the top margin of the page. This command is the equivalent of setting the First Baseline at Top Margin option in the [Typesetting dialog box](#).

### **syntax**

TypesetBaseline (State:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **State:**

This parameter specifies whether the First Baseline at Top Margin option is off or on. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Description</b>
No!	0	The first baseline is placed at the bottom of the first line of text.
Yes!	1	The first baseline is placed at the top margin.

[TypesetDlg](#)

---

## TypesetDlg

The TypesetDlg command displays the Typesetting dialog box, which lets the user set several options that affect printed text appearance. The TypesetDlg command is the equivalent of choosing Typesetting from the Layout menu.

### **syntax**

TypesetDlg ( )

TypesetBaseline

TypesetJustifyLimits

TypesetKerning

TypesetLeadingAdjust

TypesetLetterspace

TypesetUnderlineOptions

TypesetWordSpace



---

## TypesetJustifyLimits

The TypesetJustifyLimits command lets you specify the maximum and minimum amount of space between words in justified text. This command is the equivalent of setting percentages in the Word Spacing Justification Limits group box in the Typesetting dialog box.

### **syntax**

TypesetJustifyLimits (*CompressedTo:Numeric Expression; ExpandedTo:Numeric Expression*)

### **parameters**

This command has two parameters:

#### **CompressedTo:**

This numeric expression specifies the maximum compression of text in justified lines and should be an integer between 0 and 100. (The number represents the percentage of compression.) This parameter is the equivalent of setting a value in the Compressed to box in the Typesetting dialog box.

#### **ExpandedTo:**

This numeric expression specifies the maximum expansion of text in justified lines and should be an integer greater than 99. (The number represents the percentage of expansion.) This parameter is the equivalent of setting a value in the Expanded to box in the Typesetting dialog box.

TypesetDlg

---

## TypesetKerning

The TypesetKerning command specifies whether WordPerfect should automatically reduce the amount of white space between certain characters. This command is the equivalent of setting the Automatic Kerning option in the Typesetting [dialog box](#).

### **syntax**

TypesetKerning (Kerning:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **Kerning:**

This parameter specifies whether automatic kerning is on. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Description</b>
No!	0	Automatic kerning is off.
Yes!	1	Automatic kerning is on.

[KerningSpacing](#)

[TypesetDlg](#)

[TypesetManualKerningDlg](#)

---

## TypesetLeadingAdjust

The TypesetLeadingAdjust command specifies how much extra space should appear between lines. The TypesetLeadingAdjust command is the equivalent of entering measurements in the Line Height (Leading) Adjustment group box in the Typesetting dialog box.

### **syntax**

TypesetLeadingAdjust (*BetweenLines:Measurement Expression*;  
*BetweenParagraphs:Measurement Expression*)

### **parameters**

This command has two parameters:

#### **BetweenLines:**

This measurement expression specifies the amount of extra space to insert between each line in a paragraph.

#### **BetweenParagraphs:**

This measurement expression specifies the amount of extra space to insert between paragraphs.

TypesetDlg

---

## TypesetLetterspace

The TypesetLetterspace command specifies the amount of space that should be between characters. This command is the equivalent of setting an option in the Letterspacing group box in the Typesetting dialog box.

### **syntax**

TypesetLetterspace (Mode:*Value Set Member*; Percent:*Numeric Expression*)

### **parameters**

This command has two parameters:

#### **Mode:**

This parameter sets letterspacing to Normal, WordPerfect Optimal, or Percent of Optimal. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Normal!	0
---------	---

WPOptimal!	1
------------	---

PercentOfOptimal!	2
-------------------	---

#### **Percent:**

Use this *Numeric Expression* to specify the width of spaces between the letters in your document. The number you type should be a positive integer and should represent the percentage of WordPerfect Optimal Spacing that you would like to use. Numbers less than 100% reduce the amount of space between letters, while numbers greater than 100% increase the amount of space. Entering 100% is the same as using WordPerfect Optimal spacing.

If the Mode parameter is set to Normal! or WPOptimal!, the Percent parameter should be omitted. This *Numeric Expression* should be a positive integer.

TypesetDlg

---

## TypesetManualKerningDlg

The TypesetManualKerningDlg command displays the Manual Kerning dialog box which lets the user increase or decrease space between two characters. The TypesetManualKerningDlg command is the equivalent of choosing Typesetting from the Layout menu, then choosing Manual Kerning from the Typesetting dialog box.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

TypesetManualKerningDlg ( )

TypesetKerning

KerningSpacing

---

## TypesetUnderlineOptions

The TypesetUnderlineOptions command specifies whether spaces and tabs are underlined. This command is the equivalent of setting options in the Underline group box in the Typesetting dialog box.

### **syntax**

TypesetUnderlineOptions (Spaces:*Value Set Member*; Tabs:*Value Set Member*)

### **parameters**

This command has two parameters:

#### **Spaces: Tabs:**

These parameters specify whether tabs and spaces are underlined. The members of these value sets are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

No!	0
-----	---

Yes!	1
------	---

TypesetDlg

---

## TypesetWordSpace

The TypesetWordSpace command specifies the amount of space between words. This command is the equivalent of setting an option in the Word Spacing group box in the Typesetting dialog box.

### **syntax**

TypesetWordSpace (Mode:*Value Set Member*; Percent:*Numeric Expression*)

### **parameters**

This command has two parameters:

#### **Mode:**

This parameter sets word spacing to Normal, WordPerfect Optimal, or Percent of Optimal. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>
---------------	---------------------------

---

Normal!	0
---------	---

WPOptimal!	1
------------	---

PercentOfOptimal!	2
-------------------	---

#### **Percent:**

Use this *Numeric Expression* to create a space width of your own. The number you type should be a positive integer and should represent the percentage of WordPerfect Optimal Spacing that you would like to use. Numbers less than 100% reduce the amount of space between words, while numbers greater than 100% increase the amount of space. Entering 100% is the same as using WordPerfect Optimal spacing.

If the Mode parameter is set to Normal! or WPOptimal!, the Percent parameter should be omitted. This *Numeric Expression* should be a positive integer.

TypesetDlg

---

## Undelete

Use the Undelete command to restore deleted text. WordPerfect "remembers" the three most recent deletions; any of these may be restored with the Undelete command. This command is the equivalent of selecting a deletion to restore, then choosing Restore in the Undelete [dialog box](#).

### **syntax**

Undelete (WhichBuffer:*Value Set Member*)

### **parameters**

This command has one parameter:

#### **WhichBuffer:**

This parameter specifies which of the three most recent deletions is restored. The valid values are:

<b>Member</b>	<b>Numeric Equivalent</b>	<b>Definition</b>
BufferOne!	1	The most recently deleted item is restored.
BufferTwo!	2	The second most recently deleted item is restored.
BufferThree!	3	The third most recently deleted item is restored.

[UndeleteDlg](#)



---

## UndeleteDlg

The UndeleteDlg command displays the Undelete dialog box, which lets the user restore any of the three most recently deleted items. This command is the equivalent of choosing Undelete from the Edit menu.

### **syntax**

UndeleteDlg ( )

Undelete

---

## Undo

The Undo command reverses the last change made to the active document. This command is the equivalent of choosing Undo from the Edit menu.

### **syntax**

Undo ( )

---

## UNTIL

The UNTIL command is the closing command in the REPEAT-UNTIL conditional loop. The loop continues until the Test parameter in the UNTIL command is true. For example, a macro that counts to ten could use the REPEAT/UNTIL loop as follows:

```
ASSIGN (Var01; 1)
```

```
REPEAT
```

```
ASSIGN (Var01; Var01 + 1)
```

```
UNTIL (Var01 = 10)
```

In this macro, the third line adds 1 to the value of Var01 each time the loop repeats. When Var01 has the value 10, the UNTIL command is true. At that time, the macro stops repeating the loop and continues with the command immediately below the UNTIL command.

Since the condition is evaluated at the end of the loop, a Repeat-Until loop will always execute at least once, regardless of whether the condition is true or false. For information on conditional loops whose test expressions are evaluated at the top of the loop, see WHILE and ENDWHILE.

### **syntax**

UNTIL (*Test*)

### **parameters**

This command has one parameter:

#### **Test**

This parameter is a relational expression.

REPEAT

Show Me!

---

## UserBoxCaptionEdit

The UserBoxCaptionEdit command opens the Caption Editor for a specified user box, allowing subsequent commands to edit the caption text. If a user box is selected when this command is encountered, the Caption Editor opens for that box. If no user box is selected when the UserBoxCaptionEdit command is encountered, the macro pauses and the Edit User Box Caption dialog box is displayed, which lets the user specify which user box's caption to edit. Once the user specifies a user box, the User Caption Editor opens, and the macro resumes. You can close the Caption Editor with the Close command.

This command is not recordable. To use the UserBoxCaptionEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing User Box from the Graphics menu, then choosing Caption, the macro facility records the BoxEditCaption ( ) command, which performs the same function.

### **syntax**

UserBoxCaptionEdit ( )

Close

BoxEditCaption

---

## UserBoxCreate

The UserBoxCreate command creates a user box and opens the Text Box Editor. The Text Box Editor can be closed with the Close command.

This command is not recordable. To use the UserBoxCreate command, you must add it to the macro by editing the macro.

### **syntax**

UserBoxCreate ( )

Close

UserBoxEdit

---

## UserBoxCreateDlg

The UserBoxCreateDlg command displays the Select Editor dialog box, which lets the user specify whether to create a user box in the Figure, Text, or Equation Editor. The UserBoxCreateDlg command is the equivalent of choosing User Box from the Graphics menu, then choosing Create.

If you select the Dialog Box Invoke check box in this dialog box when recording a macro, no additional commands are recorded until you return to the document window. When a macro encounters this command, the macro pauses until the user returns to the document window. For other information on using dialog boxes with macros, see Dialog Boxes and Macros under Using Product Commands in Using Macros.

### **syntax**

UserBoxCreateDlg ( )

UserBoxEdit

---

## UserBoxEdit

The UserBoxEdit command opens the appropriate editor based on the contents of the user box. If a user box is selected when this command is encountered, the appropriate editor opens with that box. If no user box is selected when the UserBoxEdit command is encountered, the macro pauses and the Edit User Box dialog box is displayed, which lets the user specify which user box to edit. Once the user specifies a user box to edit, that box is brought into the appropriate editor, and the macro resumes.

This command is not recordable. To use the UserBoxEdit command, you must add it to the macro by editing the macro. Although this command is the equivalent of choosing User Box from the Graphics menu, then choosing Edit, the macro facility records the BoxEditor (Type:UserBox!) command, which performs the same function.

### **syntax**

UserBoxEdit ( )

Close

---

## UserBoxNewNumberDlg

The UserBoxNewNumberDlg command displays the User Box Number dialog box, which lets the user set a new number from which user box numbers will increment. This command is the equivalent of choosing User Box from the Graphics menu, then choosing New Number.

### **syntax**

UserBoxNewNumberDlg ( )

BoxNewNumber



---

## UserBoxOptionsDlg

The UserBoxOptionsDlg command displays the User Box Options dialog box, which lets the user specify borders, spacing, captions, and other options for user boxes. This command is the equivalent of choosing User Box from the Graphics menu, then choosing Options.

### **syntax**

UserBoxOptionsDlg ( )

BoxOptions

---

## UserBoxPositionDlg

The UserBoxPositionDlg command displays the Box Position and Size dialog box, which lets the user set the position, size, anchor, and other options for a specified User box. This command can open the Position and Size dialog box in several ways:

If a user box is being edited, the UserBoxPositionDlg command is the equivalent of choosing Box Position from the File menu in the Figure, Text, or Equation Editor.

If a user box is currently selected when the User BoxPositionDlg command is encountered, the Position and Size dialog box is displayed and the user can then set position and other options for that box.

If no box is selected, the Box Position and Size dialog box appears. This lets the user specify a number in the User Box Number text box indicating which box to modify. When the user has specified a number, the Box Position and Size dialog box is displayed, which lets the user set position and size options for that box.

### **syntax**

UserBoxPositionDlg ( )

BoxSelect

---

## UserFunction

The UserFunction command tells a "third-party" program (a program designed to work with WordPerfect) to perform an action. The action performed by the third-party program is determined by the Action parameter in the UserFunction command.

### **syntax**

UserFunction (Action:*Character Expression*)

### **parameters**

This command has one parameter:

#### **Action:**

Third-party programs look at this character expression to see if it matches a corresponding character expression they have specified. If a match occurs, the third-party program will perform the corresponding function when that program is started.

The third-party program should examine the Action string to determine whether it is for that third-party application or for another application. The Action character expression should have the following form:

**<signature>:<action>**

The <signature> string is the four character identification from the third-party application. The <action> string is any character string that the third-party application understands.

### Macro Substitution

---

## VertLineCreateDlg

The VertLineCreateDlg command displays the Create Vertical Line dialog box, which lets the user create a vertical line in the active document, specifying the line size, position, and shading. The VertLineCreateDlg command is the equivalent of choosing Line from the Graphics menu, then choosing Vertical.

### **syntax**

VertLineCreateDlg ( )

GraphicLine

HorizLineCreateDlg

HorizLineEditDlg

VertLineEditDlg

---

## VertLineEditDlg

The VertLineEditDlg command displays the Edit Vertical Line dialog box, which lets the user edit a selected vertical line. If a vertical line is already selected when the VertLineEditDlg command is encountered, the dialog box lets the user edit that line. If a vertical line is not selected when this command is encountered, WordPerfect searches backwards from the insertion point until it finds a vertical line. If it finds a vertical line code, that line is edited in the Edit Vertical Line dialog box. If no vertical line is found when searching backward, WordPerfect edits the first vertical line in the document. If no vertical line is found in any direction, this command terminates the macro or redirects macro execution to the LABEL specified by an ONERROR command.

The VertLineEditDlg command is the equivalent of choosing Line from the Graphics menu, then choosing Edit Vertical.

### **syntax**

VertLineEditDlg ( )

LABEL

ONERROR

GraphicLine

GraphicLineSelect

HorizLineCreateDlg

HorizLineEditDlg

VertLineCreateDlg

---

## **WAIT**

This command stops the macro for a specified amount of time. If the user begins typing after a macro has encountered a WAIT command, the keystrokes are ignored until the specified time has elapsed.

The length of the wait is specified in the Tenths of a Second parameter. This parameter measures time in tenths of a second, with a maximum of 600 (1 minute).

### **syntax**

WAIT (*Tenths of a Second*)

### **parameters**

This command has one parameter:

#### **Tenths of a Second**

This may be any numeric expression between 0 and 600.

[Show Me!](#)

---

## WHILE

The WHILE command marks the beginning of a WHILE-ENDWHILE conditional loop. This loop runs all of the commands between the WHILE command and the ENDWHILE command until the Test expression becomes false.

The Test expression is evaluated at the top of the WHILE-ENDWHILE loop. If the expression is false, the WHILE-ENDWHILE loop will not execute, and macro execution resumes with the command following the ENDWHILE command. For information on conditional loops whose text expressions are evaluated at the end of the loop, see REPEAT and UNTIL.

### **syntax**

WHILE (*Test*)

### **parameters**

This command has one parameter:

#### **Test**

This parameter is a relational expression.

ENDWHILE

Show Me!

---

## WindowCascade

The WindowCascade command "cascades" all document windows, so that the document windows overlap, with the title bar of each document remaining visible. The WindowCascade command is the equivalent of choosing Cascade from the Window menu.

### **syntax**

WindowCascade ( )

WindowTile



---

## WindowTile

The WindowTile command "tiles" all document windows, so that the open document windows each take an approximately equal amount of space in the WordPerfect application area. The documents do not overlap. The WindowTile command is the equivalent of choosing Tile from the Window menu.

### **syntax**

WindowTile ( )

WindowCascade

---

## WordCountDlg

The WordCountDlg command activates the Word Count dialog box, which displays the number of words in the active document. The WordCountDlg command is the equivalent of choosing Word Count from the Tools menu.

### **syntax**

WordCountDlg ( )

---

## WPCharactersDlg

The WPCharactersDlg command displays the WordPerfect Characters dialog box, which lets the user access all WordPerfect Corporation extended characters. This command is the equivalent of choosing WP Characters from the Font menu or pressing Ctrl+w.

This command is not recordable. To use this command, you must add it to the macro by editing the macro.

### **syntax**

WPCharactersDlg ( )

---

## Zoom

This command allows you to "zoom in" on a page; to enlarge the display size of the page to a percent specified in the Percent parameter.

### **syntax**

Zoom (Percent:*Numeric Expression*)

### **parameters**

This command has one parameter:

#### **Percent**

This is a numeric expression.

Zoom100

Zoom150

Zoom200

Zoom50

Zoom75

ZoomDlg

ZoomToPageWidth

---

## **Zoom100**

Zoom100 will return the display size of the page to its original size (100 %). 100 percent is the default size of a document on the screen. This is the equivalent of choosing 100% from the Zoom submenu under the View menu

### **syntax**

Zoom100 ( )

Zoom150

Zoom200

Zoom50

Zoom75

ZoomDlg

ZoomToPageWidth

---

## **Zoom150**

This command allows you to "zoom in" on the active document. The document will be displayed at 150% of its original size. This is the equivalent of choosing 150% from the Zoom submenu under the View menu

### **syntax**

Zoom150 ( )

Zoom100

Zoom200

Zoom50

Zoom75

ZoomDlg

ZoomToPageWidth

---

## **Zoom200**

This command allows you to "zoom in" on the active document. The document will be displayed at 200% of its original size. This is the equivalent of choosing 200% from the Zoom submenu under the View menu.

### **syntax**

Zoom200 ( )

Zoom100

Zoom150

Zoom50

Zoom75

ZoomDlg

ZoomToPageWidth

---

## Zoom50

Zoom50 allows you to view the active document at 50% of its original size. This is the equivalent of choosing 50% from the Zoom submenu under the View menu.

### **syntax**

Zoom50 ( )

Zoom100

Zoom150

Zoom200

Zoom75

ZoomDlg

ZoomToPageWidth



---

## Zoom75

Zoom75 allows you to view the active document at 75% of its original size. This is the equivalent of choosing 75% from the Zoom submenu under the View menu

### **syntax**

Zoom75 ( )

Zoom100

Zoom150

Zoom200

Zoom50

ZoomDlg

ZoomToPageWidth

---

## **ZoomDlg**

The ZoomDlg command displays the Zoom Dialog box which allows you to specify the percentage you would like to "zoom in" or "zoom out" on the active document.

This command is not recordable. You must insert it while editing the macro.

### **syntax**

ZoomDlg ( )

Zoom100

Zoom150

Zoom200

Zoom50

Zoom75

ZoomToPageWidth

---

## **ZoomToPageWidth**

The ZoomToPageWidth command allows you to view the active document in a size that shows all the text between the page's left and right margins.

### **syntax**

ZoomToPageWidth ( )

Zoom100

Zoom150

Zoom200

Zoom50

Zoom75

ZoomDlg

---

## User-defined Dialog Boxes

One of the most powerful features in WordPerfect macros are dialog boxes. Dialog boxes are easy ways to communicate with users to share or solicit information. Of all the ways to create a dialog box, the most versatile and convenient is the User-defined Dialog Box. The commands for the User-defined Dialog Box allow you to create dialog boxes as easily as you use any other programming command in WordPerfect. The steps taken to create a dialog box are:

Define the dialog

Add any wanted controls (options):

CheckBox  
ColorWheel  
ComboBox  
Counter  
EditBox  
FilenameBox  
Frame  
GroupBox  
HorizontalLine  
HotSpot  
Icon  
ListBox  
ListItem  
PopUpButton  
PushButton  
RadioButton  
ScrollBar  
Text  
VerticalLine  
Viewer

Display the dialog

Destroy the dialog

Any dialog boxes that are created must have their own unique identity to distinguish them from other dialog boxes you use during the course of your macro. That is the purpose of the **DialogDefine** command. Then, when you create a dialog box, you have the options to decide where in the document window you would like it to appear, which check boxes, buttons, or other controls you would like to include in the dialog box, and how you would like it to look. The options under "Add any wanted controls" above are those you will use to specify the look of your dialog boxes and to add any check boxes or buttons you choose to use. The **DialogDisplay** command tells which dialog box to display and **DialogDestroy** tells your macro to release all links and destroy the dialog box.

You will notice that there is a slight difference in syntax between other programming commands and the User-defined Dialog Box commands. In other programming commands

the syntax uses the parameter name (such as "WP" for the parameter "Product Prefix" in the example below). In the User-defined Dialog Box commands, you use the value of the parameter. For example, in the APPLICATION command, the syntax is:

Application (Product Prefix; Product Identifier; DEFAULT; Filename)

so you would write the command:

Application (WP;WPWP;DEFAULT).

The parameter, "Product Prefix" uses the parameter name, "WP". In the User-defined Dialog Box commands, each parameter will have values, numbers, that will be used instead of parameter names. For example, in the DialogDefine command, the syntax is:

DialogDefine(DialogId; HPos; VPos; Width; Height; Style; Title)

Each of these parameters has a numeric value, a number, that you use to indicate your specification of that parameter. So you would write the command:

DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Test Dialog").

For further information, please see:

[ControllD Parameter](#)

[MacroDialogResult](#)

*Note on Style Parameters:* In many of the **DialogAdd...** commands, there will be a parameter that specifies a style with a value and a definition. If there is a default style that is not listed, it can be specified with the value of "0"; this means the style that is the opposite of that which is listed.

## ControlId Parameter

The ControlId parameter is used to set the focus (or default, where applicable) on a specific dialog control item. A dialog control (option) will be displayed with the focus if the ControlId parameter of both the control option and the DialogDisplay command are the same .

Valid numbers for the ControlId parameter are 0, 1, 2, and anything greater than 10. The numbers 1 and 2 are used for OK and Cancel push buttons. If 1 or 2 are used, they will turn off the 1 and/or 2 (OK and Cancel) style(s) of the DialogDefine command.

The value of the control (from the ControlId parameter) is assigned to the variable MacroDialogResult. Once the dialog is dismissed, the value of the control used to dismiss the dialog remains in the variable MacroDialogResult. This variable can be tested to see which control dismissed the dialog.

If the ControlId is specified as the number 0 or a string of text, a number greater than 10 is assigned to MacroDialogResult when the **DialogAdd...** command is processed.

## MacroDialogResult

Once a dialog box is dismissed, the variable MacroDialogResult will contain a number. The number is determined by the component of the dialog box (the dialog control) that was used to dismiss the box (for example, an OK or a Cancel button). If you do not define the number to be returned when you set the parameters for a control, then the default dialog box numbers will apply. The default dialog box numbers are 1 if the OK button was pressed and 2 if the Cancel button was pressed. If you wish to test to see what number this variable contains, you must do so before the DialogDestroy command is sent. Once the DialogDestroy command takes effect, the MacroDialogResult variable will be cleared.

### example:

//In this example the default or selected item will be the edit box.

//When the dialog is dismissed, the program assigns the value of the control used (OK or CANCEL) to dismiss the dialog box to the variable Result.

```
Application (WP;WPWP;Default;"WPWPUS.WCD")
```

```
DialogDefine(1000;50;50;150;155;1+2+16;"Test Dialog")
```

```
DialogAddEditBox(1000;100;10;10;100;13;1;editvar;10)
```

```
DialogAddRadioButton(1000;101;10;30;100;10;"Select this item";radiovar)
```

```
DialogDisplay(1000;100)
```

```
Result:=MacroDialogResult
```

```
DialogDestroy(1000)
```

---

## DialogDefine

The DialogDefine command gives the dialog a specific identification (id) and specifies the position, size, function and look of the dialog. Each dialog box must have a unique id so that when other options, such as radio buttons or scroll bars, are specified, the macro facility will know which dialog box gets which options. This id can be either a numeric or character expression, so long as you remember the id for reference later in such commands as DialogDisplay and DialogDestroy.

### syntax

DialogDefine(DialogId; HPos; VPos; Width; Height; Style; Title)

### parameters:

#### DialogId

The DialogId parameter defines the id of the dialog box. The id may either be a numeric or character expression, but all subsequent dialog commands that affect this dialog box must contain the same id as assigned here.

#### HPos

The HPos parameter sets the horizontal position of the dialog box in the document window in either dialog units or percentage of the screen from the left side of the screen (see Percent under the Style parameter).

#### VPos

The VPos parameter sets the vertical position of the dialog box in the document window in either dialog units or percentage of the screen from the top of the screen (see Percent under the Style parameter).

#### Width

The Width parameter sets the width of the dialog box in dialog units.

#### Height

The Height parameter sets the height of the dialog box in dialog units.

#### Style

The Style parameter sets the options to specify the appearance of the dialog box. Use the number listed under Numeric Equivalent to create the option listed under Style. You may use more than one option by putting a "+" in between the options. For example, if you want dialog box buttons for OK and Cancel and you wish the dialog to be placed by percentage in the middle of the screen, you would have

1+2+16

as the numbers for the Style parameter. The styles, values and definitions for this parameter are:

Style	Value	Definition
OK	1	Puts an "OK" button on the dialog box.
Cancel	2	Puts a "Cancel" button on the dialog box.
Modeless	8	Dialog box does not need to be dismissed before doing other tasks. If this number isn't added to



the style, the dialog box must be dismissed before doing other tasks.

Percent	16	Sets measurement to percent of the screen. This will cause the values in the HPos and VPos parameters to become percentages of the screen size rather than fixed numbers. For example, if this number is set, setting the HPos and VPos values to 50 and 50 will cause the dialog box to display in the center of the screen.
NoFrame	32	Takes the frame off the dialog box.
Sizeable	128	Allows re-sizing of the dialog box.
NoTitle	256	Takes the Title bar off the dialog box.

### **Title**

The Title parameter specifies the text appearing on the title bar .

[DialogAddCheckBox](#)

[DialogAddColorWheel](#)

[DialogAddComboBox](#)

[DialogAddCounter](#)

[DialogAddEditBox](#)

[DialogAddFilenameBox](#)

[DialogAddFrame](#)

[DialogAddGroupBox](#)

[DialogAddHLine](#)

[DialogAddVLine](#)

[DialogAddHotSpot](#)

[DialogAddIcon](#)

[DialogAddListBox](#)

[DialogAddListItem](#)

[DialogAddPopUpButton](#)

[DialogAddPushButton](#)

[DialogAddRadioButton](#)

[DialogAddScrollBar](#)

[DialogAddText](#)

[DialogAddViewer](#)

[ShowMe!](#)

---

## DialogDisplay

The DialogDisplay command will display the dialog box specified in DialogId parameter. The DialogDisplay command also allows you to indicate which button or control option you would like as the default (the one which will be activated if the user presses **Enter** instead of clicking on a choice).

### **syntax**

DialogDisplay(DialogId; ControlId)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box to be displayed. This id must match the id in the DialogId parameter of the DialogDefine command.

#### **ControlId**

The ControlId parameter specifies the name or number of the button or control option that will become the default when the dialog box is displayed. There can be only one default option in each dialog box.

DialogDefine

DialogDestroy

ShowMe!

---

## DialogDestroy

The DialogDestroy command will purge from memory the dialog box specified in the DialogId parameter. You must destroy a dialog box before creating another of the same name.

### **syntax**

DialogDestroy(DialogId)

### **parameter:**

#### **DialogId**

This parameter specifies the name or number of the dialog box to be destroyed. This id must match that of a dialog box previously created with the DialogDefine command.

DialogDefine

DialogDisplay

ShowMe!

---

## DialogAddCheckBox

The DialogAddCheckBox command positions a check box in the dialog box and puts text to the right of the check box.

### **syntax**

DialogAddCheckBox(DialogId; ControlId; HPos; VPos; Width; Height; Text; Variable)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the check box or boxes will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the check box that will become the default when the dialog box is displayed. There can be only one default option in each dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the check box, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the check box, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the check box in dialog units.

#### **Height**

The Height parameter specifies the height of the check box in dialog units.

#### **Text**

The Text parameter contains the character string that appears as the text next to the check box.

#### **Variable**

The Variable parameter holds the return value of the check box. Before the macro runs, the variable needs to be initialized (set to 0). Once the box is checked, this variable will be set to 1; if it is not checked, this variable will remain at 0.

DialogDefine

ShowMe!

---

## DialogAddColorWheel

The DialogAddColorWheel command defines a color wheel control in the dialog box and returns a number representing the color chosen.

### **syntax**

DialogAddColorWheel(DialogId; ControlId; HPos; VPos; Width; Height; Variable)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the color wheel will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the color wheel that will become the default when the dialog box is displayed. There can be only one default option in each dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the color wheel, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the color wheel, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the color wheel in dialog units.

#### **Height**

The Height parameter specifies the height of the color wheel in dialog units.

#### **Variable**

The Variable parameter holds the return value of the color wheel. This returns a number that must be broken into its Red, Green, and Blue properties using these formulas. Red = Return % 256, Green = Return / 256 % 256, Blue = Return / 256 / 256. Before the macro runs, the variable needs to be initialized (set to 0).

[DialogDefine](#)

[ShowMe!](#)

---

## DialogAddComboBox

The DialogAddComboBox command creates an edit box/drop-down list box containing items defined by DialogAddListItem.

### **syntax**

DialogAddComboBox(DialogId; ControlId; HPos; VPos; Width; Height; Style; Variable; DataLimit)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the combo box will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the combo box; this control id must be the same as that specified by the list items which are to go in the combo box.

#### **HPos**

The HPos parameter specifies the horizontal position of the combo box, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the combo box, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the combo box in dialog units.

#### **Height**

The Height parameter specifies the height of the combo box in dialog units.

#### **Style**

The Style parameter specifies the options for dialog box. The styles, values and definitions are:

<b>Style</b>	<b>Value</b>	<b>Definition</b>
Simple	0	Appears as an edit box with an opened drop-down list.
Dropdown	8	Appears as an edit box with a drop-down list.
Droplist	16	Appears as a drop-down list that has a 3D border.

#### **Variable**

The Variable parameter holds the return value of the combo box. The list item chosen or text in the edit box will be the return value as a character expression.

#### **DataLimit**

The DataLimit parameter defines the limit to the number of characters allowed in the edit box.

DialogDefine

DialogAddListItem

ShowMe!

---

## DialogAddCounter

The DialogAddCounter command defines a counter control that will accept numbers between a preset minimum and maximum value. This option also has increment and decrement arrows.

### syntax

DialogAddCounter(DialogId; ControlId; HPos; VPos; Width; Height; Format; Variable; MinValue; MaxValue; StepValue)

### parameters:

#### DialogId

The DialogId parameter specifies the name or number of the dialog box in which the counter will be placed.

#### ControlId

The ControlId parameter specifies the name or number of the counter that will become the default when the dialog box is displayed. There can be only one default option in each dialog box.

#### HPos

The HPos parameter specifies the horizontal position of the counter, in dialog units, from the left side of the dialog box.

#### VPos

The VPos parameter specifies the vertical position of the counter, in dialog units, from the top of the dialog box.

#### Width

The Width parameter specifies the width of the counter in dialog units.

#### Height

The Height parameter specifies the height of the counter in dialog units.

#### Format

The Format parameter defines the unit of measure to be used with counter. The format definitions are:

Format	Value	Definition
	0	Normal
"w"	1	WP units.
"p"	2	Points.
"c"	3	Centimeters.
".0""	4	Inches with a decimal place.
"%"	5	Percent.
	6	Inches.
"j"	7	Inches.



**Variable**

The Variable parameter holds the return value of the counter. The number entered in the counter will be returned in this variable. Any number contained in this variable before displaying the dialog will be the initial number shown in the counter control. Note that by assigning this variable a measurement, the result will be returned in the same unit of measure.

**MinValue**

The MinValue parameter specifies the lowest, or minimum, value of the counter.

**MaxValue**

The MaxValue parameter specifies the highest, or maximum, value of the counter.

**StepValue**

The StepValue parameter defines what the increment value will be if you use the increment arrows.

DialogDefine

ShowMe!

---

## DialogAddEditBox

The DialogAddEditBox command defines an edit box within the dialog box in which the user can type text.

### **syntax**

DialogAddEditBox(DialogId; ControlId; HPos; VPos; Width; Height; Style; Variable; DataLimit)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the edit box will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the edit box that will become the default when the dialog box is displayed. There can be only one default option in each dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the edit box, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the edit box, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the edit box in dialog units.

#### **Height**

The Height parameter specifies the height of the edit box in dialog units.

#### **Style**

The Style parameter defines the options for the edit box. The styles, values and definitions are:

<b>Style</b>	<b>Value</b>	<b>Definition</b>
Left	1	Left justified.
Right:	2	Right justified when not used in conjunction with WPChars.
Center	4	Center justified when not used in conjunction with WPChars.
VScroll	8	Adds vertical <u>scroll bar</u> .
HScroll	16	Adds horizontal scroll bar.
WPChars	32	Accepts WordPerfect Characters.
Multiline	64	Multiple line ability. See also WordWrap style option.
Uppercase	128	Exclusively uppercase.
Lowercase	256	Exclusively lowercase.

Password	512	Types "*" instead of an actual character.
WordWrap	1024	Wraps line if string exceeds the width of the edit box.
SoftRet	2048	SRT's are formatted into the output.
Attributes	4096	Allows attributes (bold, underline, etc.).
NoTabs	8192	No tabs will be exported with output.
NoWPChar	16384	WordPerfect character box is not accessible.

### **Variable**

The Variable parameter holds the return value of the edit box. The return contains the text last present in the edit box. Any string contained in this variable before displaying the dialog will be the initial string shown in the edit box.

### **DataLimit**

The DataLimit parameter sets a limit to the number of characters allowed in the edit box.

### DialogDefine

### ShowMe!

---

## DialogAddFileNameBox

The DialogAddFileNameBox defines a filename box in the dialog box. This is an edit box which allows you to bring up a file from the directory specified in the DefDir parameter. The file type (.bat, .doc, etc.) is specified in the Template parameter. If you wish to have a filename listed when the Filename box first appears, you can define the name to appear in a variable as shown in the example for this command.

### **syntax**

DialogAddFileNameBox(DialogId; ControlId; HPos; VPos; Width; Height; Style; Variable; DefDir; Template)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the filename box will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the filename box that will become the default when the dialog box is displayed. There can be only one default option in each dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the filename box, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the filename box, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the filename box in dialog units.

#### **Height**

The Height parameter specifies the height of the filename box in dialog units.

#### **Style**

The Style parameter specifies the options for dialog box. The styles, values and definitions are:

<b>Style</b>	<b>Value</b>	<b>Definition</b>
DirOnly	1	Only directories may be selected.
NoValidate	2	Does no error checking to see if directory exists when used in conjunction with DirOnly.

#### **Variable**

The Variable parameter holds the return value of the filename box.

#### **DefDir**

The DefDir parameter defines the directory which will become the default directory in the filename box.

**Template**

The Template parameter designates which type of file will be brought up in the filename box (\*.\*, \*.bat, etc.).

DialogDefine

ShowMe!

---

## DialogAddFrame

The DialogAddFrame command defines a frame in the dialog box. This frame can be used in conjunction with other boxes within the dialog box. For example, you can define a frame for an edit box within the dialog box.

### **syntax**

DialogAddFrame(DialogId; ControlId; HPos; VPos; Width; Height; Style)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the frame will be placed.

#### **ControlId**

The ControlId parameter specifies a name or number for the frame in the dialog box. This name or number is used for displaying and dismissing the dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the frame, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the frame, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the frame in dialog units.

#### **Height**

The Height parameter specifies the height of the frame in dialog units.

#### **Style**

The Style parameter specifies options for the frame in the dialog box. The style, value and definitions are:

<b>Style</b>	<b>Value</b>	<b>Definition</b>
Frame	0	Displays a frame outline only.
Black	0	Color of frame.
Gray	1	Color of frame.
White	2	Color of frame.
Filled	4	Displays a black-filled frame.

DialogDefine

ShowMe!

---

## DialogAddGroupBox

The DialogAddGroupBox command defines a group box component within the dialog box. This is a useful feature in a dialog box for showing relationships among items. If, for example, you had several options in a dialog box that dealt with the address on a letter and the rest of the options dealt with the heading and content, you might group all the address components together in a group box. This would make it simple for the user to see the relationship and logically proceed through the options.

### **syntax**

DialogAddGroupBox(DialogId; ControlId; HPos; VPos; Width; Height; Title)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the group box will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the group box. This will be used for displaying and dismissing the dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the group box, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the group box, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the group box in dialog units.

#### **Height**

The Height parameter specifies the height of the group box in dialog units.

#### **Title**

The Title parameter specifies a title to be centered in the top line of the group box. The title is a character expression.

[DialogDefine](#)

[DialogAddHLine](#)

[DialogAddVLine](#)

[ShowMe!](#)

---

## DialogAddHLine

The DialogAddHLine command defines a horizontal line in the dialog box. As with the DialogAddGroupBox command, the DialogAddHLine command can be used to separate options in a dialog box into logical groupings to show relationships among options.

### **syntax**

DialogAddHLine(DialogId; ControlId; HPos; VPos; Length)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the horizontal line will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the horizontal line. This will be used for displaying and dismissing the dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the horizontal line, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the horizontal line, in dialog units, from the top of the dialog box.

#### **Length**

The Length parameter specifies the length of the horizontal line in dialog units.

[DialogDefine](#)

[DialogAddGroupBox](#)

[DialogAddVLine](#)

[ShowMe!](#)



---

## DialogAddVLine

The DialogAddVLine command defines a vertical line in the dialog box. As with the DialogAddGroupBox command, the DialogAddVLine command can be used to separate options in a dialog box into logical groupings to show relationships among options.

### **syntax**

DialogAddVLine(DialogId; ControlId; HPos; VPos; Length)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the vertical line will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the vertical line. This will be used for displaying and dismissing the dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the vertical line, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the vertical line, in dialog units, from the top of the dialog box.

#### **Length**

The Length parameter specifies the length of the vertical line in dialog units.

[DialogDefine](#)

[DialogAddGroupBox](#)

[DialogAddHLine](#)

[ShowMe!](#)

---

## DialogAddHotSpot

The DialogAddHotSpot command defines an area of the dialog box as a "hot spot"; if you click on the hot spot, it will dismiss the dialog box. A hot spot can be defined with the same HPos and VPos, Height and Width parameters as other options of the dialog box. By so doing, you can make these other options become hot spots.

### **syntax**

DialogAddHotSpot(DialogId; ControlId; HPos; VPos; Width; Height; Style)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the hot spot will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the hot spot. This will be used for displaying and dismissing the dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the hot spot, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the hot spot, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the hot spot in dialog units.

#### **Height**

The Height parameter specifies the height of the hot spot in dialog units.

#### **Style**

The Style parameter specifies options for the hot spot in the dialog box. The style, value and definition is:

<b>Style</b>	<b>Value</b>	<b>Definition</b>
Click	1	Activated by a single click.

### DialogDefine

### ShowMe!

---

## DialogAddIcon

The DialogAddIcon command defines an Icon to be displayed in the dialog box. Note: The icon must be contained in a .DLL and you must know the name of the icon in order to define the IconName and ModuleLink parameters.

### **syntax**

DialogAddIcon(DialogId; ControlId; HPos; VPos; Width; Height; IconName; ModuleLink)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the icon will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the icon. This will be used for displaying and dismissing the dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the icon, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the icon, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the icon in dialog units.

#### **Height**

The Height parameter specifies the height of the icon in dialog units.

#### **IconName**

The Icon name specifies the name of the desired icon. The icon must be contained in a .DLL and you must know the name of the icon in order to define the IconName and ModuleLink parameters.

#### **ModuleLink**

The ModuleLink parameter specifies the handle of the DLL that contains the icon. The icon must be contained in a .DLL and you must know the name of the icon in order to define the IconName and ModuleLink parameters.

DialogDefine

ShowMe!

---

## DialogAddListBox

The DialogAddListBox command defines a list box containing options defined by the DialogAddListItem command.

### **syntax**

DialogAddListBox(DialogId; ControlId; HPos; VPos; Width; Height; Style; Variable)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the list box will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the list box; this control id must be the same as that specified by the list items which are to go in the list box.

#### **HPos**

The HPos parameter specifies the horizontal position of the list box, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the list box, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the list box in dialog units.

#### **Height**

The Height parameter specifies the height of the list box in dialog units.

#### **Style**

The Style parameter specifies options for the list box in the dialog box. The style, value and definitions are:

<b>Style</b>	<b>Value</b>	<b>Definition</b>
Sort	1	Sorts the list items alphabetically
NameSearch	2	Allows name search in the list.

#### **Variable**

The Variable parameter contains the return value of the list item. The list item chosen will be the return value as a character expression.

DialogDefine

DialogAddListItem

ShowMe!

---

## DialogAddListItem

The DialogAddListItem command defines a list item to be used in the combo box, list box, or pop-up button defined in the DialogAddComboBox, DialogAddListBox, and DialogAddPopUpButton commands.

### **syntax**

DialogAddListItem(DialogId; ControlId; String)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the list item will be placed.

#### **ControlId**

The ControlId parameter contains the same name or number of the list box, combo box, or pop-up button into which the list item will be placed.

#### **String**

The String parameter defines a text string for list item.

[DialogDefine](#)

[DialogAddListBox](#)

[ShowMe!](#)

---

## DialogAddPopUpButton

The DialogAddPopUpButton command defines a pop-up list box in the dialog box containing items defined by the DialogAddListItem command.

### **syntax**

DialogAddPopUpButton(DialogId; ControlId; HPos; VPos; Width; Height; Variable)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the pop-up list box will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the pop-up list box; this control id must be the same as that specified by the list items which are to go in the pop-up list box.

#### **HPos**

The HPos parameter specifies the horizontal position of the pop-up list box, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the pop-up list box, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the pop-up list box in dialog units.

#### **Height**

The Height parameter specifies the height of the pop-up list box in dialog units.

#### **Variable**

The Variable parameter contains the return value of the list item chosen from the pop-up list box. The list item chosen will be the return value as a character expression.

[DialogDefine](#)

[DialogAddListItem](#)

[ShowMe!](#)

---

## DialogAddPushButton

The DialogAddPushButton command defines a push button in the dialog box..

### **syntax**

DialogAddPushButton(DialogId; ControlId; HPos; VPos; Width; Height; Style; Text)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the push button will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the push button that will become the default when the dialog box is displayed. There can be only one default option in each dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the push button, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the push button, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the push button in dialog units.

#### **Height**

The Height parameter specifies the height of the push button in dialog units.

#### **Style**

The Style parameter specifies options for the push button in the dialog box. The style, value and definitions are:

<b>Style</b>	<b>Value</b>	<b>Definition</b>
Default	1	Sets button as default.

#### **Text**

The Text parameter contains the text string that will appear on the push button.

#### DialogDefine

#### ShowMe!

---

## DialogAddRadioButton

The DialogAddRadioButton command defines a radio button in the dialog box.

### **syntax**

DialogAddRadioButton(DialogId; ControlId; HPos; VPos; Width; Height; Text; Variable)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the radio button will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the radio button that will become the default when the dialog box is displayed. There can be only one default option in each dialog box.

#### **HPos**

The HPos parameter specifies the horizontal position of the radio button, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the radio button, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the radio button in dialog units.

#### **Height**

The Height parameter specifies the height of the radio button in dialog units.

#### **Text**

The Text parameter defines the text string that will appear on the radio button.

#### **Variable**

Variable that holds the return value of the radio button. The string of text in the Text parameter is assigned to this variable.

DialogDefine

ShowMe!



---

## DialogAddScrollBar

The DialogAddScrollBar command defines a horizontal or vertical scroll bar in the dialog box. When you set the parameters below, you can set where the scroller appears initially in the scroll bar. You can also set the MinValue and MaxValue parameters to give the scroll bar a range and set the Variable parameter to dictate where within that range the scroller will appear.

### **syntax**

DialogAddScrollBar(DialogId; ControlId; HPos; VPos; Width; Height; Style; Variable; MinValue; MaxValue)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the scroll bar will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the scroll bar. This will be used for displaying and dismissing the dialog.

#### **HPos**

The HPos parameter specifies the horizontal position of the scroll bar, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the scroll bar, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the scroll bar in dialog units.

#### **Height**

The Height parameter specifies the height of the scroll bar in dialog units.

#### **Style**

The Style parameter specifies options for the scroll bar in the dialog box. The style, value and definitions are:

<b>Style</b>	<b>Value</b>	<b>Definition</b>
Left/Top	1	Positions the scroll box at left or top (depending on scroll bar). A number in the Variable will override this.
Right/Bot	2	Positions the scroll box at right or bottom. A number in the Variable will override this.
VScroll	8	Specifies a vertical scroll bar.
HScroll	16	Specifies a horizontal scroll bar.

#### **Variable**

The Variable parameter contains the return value of the scroll bar. This value is determined by the position of the scroller within the scroll bar. You can set the value in the variable to dictate where within the range specified by the MinValue and MaxValue parameters you would like the scroller to appear. You can also check this variable after the dialog box has been dismissed to determine the final setting of the scroller when the dialog box was dismissed.

**MinValue**

The MinValue parameter defines the lowest value of the range you specify for the scroll bar.

**MaxValue**

The MaxValue parameter defines the highest value of the range you specify for the scroll bar.

DialogDefine

ShowMe!

---

## DialogAddText

The DialogAddText command allows you to insert text into the dialog box. This text can be set in a visible recessed box or shadow box, or it can simply display right, center, or left justified in a non-visible box within the dialog box. In the HPos, VPos, Width, and Height parameters, the dialog units will specify the box position, regardless of whether the text box is visible or not.

### syntax

DialogAddText(DialogId; ControlId; HPos; VPos; Width; Height; Style; Text)

### parameters:

#### DialogId

The DialogId parameter specifies the name or number of the dialog box in which the text will be displayed.

#### ControlId

The ControlId parameter specifies the name or number of the text. This will be used for displaying and dismissing the dialog.

#### HPos

The HPos parameter specifies the horizontal position of the text box, in dialog units, from the left side of the dialog box.

#### VPos

The VPos parameter specifies the vertical position of the text box, in dialog units, from the top of the dialog box.

#### Width

The Width parameter specifies the width of the text box in dialog units.

#### Height

The Height parameter specifies the height of the text box in dialog units..

#### Style

The Style parameter specifies options for the text box in the dialog box. The style, value and definitions are:

Style	Value	Definition
Left	1	Left justified.
Right	2	Right justified when not used in conjunction with WPChars.
Center	4	Center justified when not used in conjunction with WPChars.
RecessBox	8	Text displayed with recessed background.
ShadowBox	16	Text displayed with shadowed background.
WPChars	32	WordPerfect characters.

#### Text

The Text parameter is a character expression that defines the text to be displayed in the text box.

DialogDefine

ShowMe!

---

## DialogAddViewer

The DialogAddViewer command defines a mini-window in the dialog box that displays a file. This "mini-window" has horizontal and vertical scroll bars attached at the bottom and right sides of the window; all you do is define the positioning and size of the window and tell WordPerfect the name of the file to display.

### **syntax**

DialogAddViewer(DialogId; ControlId; HPos; VPos; Width; Height; Filename)

### **parameters:**

#### **DialogId**

The DialogId parameter specifies the name or number of the dialog box in which the viewer will be placed.

#### **ControlId**

The ControlId parameter specifies the name or number of the viewer. This will be used for displaying and dismissing the dialog.

#### **HPos**

The HPos parameter specifies the horizontal position of the viewer, in dialog units, from the left side of the dialog box.

#### **VPos**

The VPos parameter specifies the vertical position of the viewer, in dialog units, from the top of the dialog box.

#### **Width**

The Width parameter specifies the width of the viewer in dialog units.

#### **Height**

The Height parameter specifies the height of the viewer in dialog units.

#### **Filename**

This character expression specifies the name of file to be viewed in the mini-window.

DialogDefine

ShowMe!

---

## Example for APPLICATION

APPLICATION (WP;WPWP;DEFAULT)

Type("Please type your name and the date below, then begin your essay.")

HardReturn()

HardReturn()

---

## Example for ASSERTCANCEL and ELSE

```
APPLICATION (WP;WPWP;Default)
ONCANCEL CALL (End@)
LABEL (DocName@)
GetWPData (Name;Name!)
IF (Name <> "")
PosDocBottom ()
HardReturn ()
HardReturn ()
Type (Name)
QUIT
ELSE
ASSERTCANCEL
ENDIF
LABEL (End@)
PROMPT ("NO NAME";"This file has no name";3;;)
WAIT (30)
ENDPROMPT
FileSaveAsDlg ()
RETURN
```

---

## Example for ASSIGN

APPLICATION (WP;WPWP;Default)

ASSIGN (Again;"Y")

ASSIGN (Create;"CREATING MERGE SECONDARY FILE")

WHILE ((Again="Y") OR (Again="y"))

FOREACH (Item;{"Name";"Street";"City";"State";"Zip";"Telephone"})

GETSTRING (X;"Enter Customer's " + Item + " .";Create)

Type (X)

MergeEndField ()

ENDFOR

MergeEndRecord ()

GETSTRING (Again;Length=1;"Create Another Record?";Create)

ENDWHILE



---

## Example for BEEP

APPLICATION (WP;WPWP;DEFAULT)

BEEP

DEFAULTUNITS (Inches)

GETUNITS(width;"Enter width of the text box (in inches)";"WIDTH")

BEEP

GETUNITS(height;"Enter height of the text box (in inches)";"HEIGHT")

TextBoxCreate()

BoxPosition (AutoMode:SetBoth!; Width:width; Height:height)

MacroStatusPrompt (On!;"Type Information -- Press Ctrl-F4 to Exit.")

PauseKey (Close!)

Close (Yes!)

---

## Example for BYTELEN

GETSTRING (VarName;"Type your first name ";"NAME")

BYTELEN (LenOfVar;VarName)

NUMSTR (CharLen;0;LenOfVar)

PROMPT ("Name Length";VarName + " is " + CharLen + " bytes long. ";3;;)

PAUSE

ENDPROMPT

---

## Example for BYTEPOS

```
APPLICATION (WP;WPWP;Default;"WPWPUS.WCD")
GETSTRING(Origexp;"Enter the original expression";"BytePos Macro")
GETSTRING(Subexp;"Enter the sub expression";"BytePos Macro")
BYTEPOS(Byteenum;Subexp;Origexp)
IF(Byteenum=0)
PROMPT("BytePos Macro";"The sub expression "" + Subexp + "" was not found in the
original expression.";3;;)
ELSE
NUMSTR(Bytestr;;Byteenum)
PROMPT("BytePos Macro";"The sub expression"" + Subexp + "" begins at byte " + Bytestr
+ ".";3;;)
ENDIF
PAUSE
ENDPROMPT
```

---

## Example for CALL

APPLICATION (WP;WPWP;Default)

Type("To: ")

ASSIGN(Memo;"to")

CALL (Input@)

Type("From: ")

ASSIGN(Memo;"from")

CALL(Input@)

QUIT

LABEL(Input@)

GETSTRING(Text;"Who is this " + Memo + "?: ";"User Input")

Type(Text)

HardReturn()

RETURN

---

## Example for CANCELON and CANCELOFF

```
APPLICATION (WP;WPWP;Default)
CANCELON
ONCANCEL (Random@)
LABEL (Top@)
FOR (Answer;0;Answer<99;Answer + 1)
PROMPT ("Random Number";
"Press [ESC] to select a random number";;)
ENDFOR
GO (Top@)
LABEL (Random@)
CANCELOFF
NUMSTR (Number;0;Answer)
PROMPT("Random Number";"The random number you chose was: " +
Number;;;)
WAIT(50)
ENDPROMPT
```

---

## Example for CASE and MENU

```
APPLICATION(WP;WPWP;Default)
PageMargins(Left:4")
Type(Text:"Thank You,")
HardReturn()
HardReturn()
HardReturn()
HardReturn()
MENU(NAME;DIGIT;;;{"Attorney 1";"Attorney 2";"Attorney 3"})
CASE(NAME;{1;Atty1@;2;Atty2@};Atty3@)
LABEL(Atty1@)
Type(Text:"Attorney 1")
HardReturn()
Type(Text:"Attorney at Law")
QUIT
LABEL(Atty2@)
Type(Text:"Attorney 2")
HardReturn()
Type(Text:"Attorney at Law")
QUIT
LABEL(Atty3@)
Type(Text:"Attorney 3")
HardReturn()
Type(Text:"Attorney at Law")
QUIT
```

---

## Example for CHAIN

```
APPLICATION (WP;WPWP;DEFAULT)
```

```
MergeExecute("test.pri";"test.sec")
```

```
PosDocVeryTop()
```

```
CHAIN("c:\WPWin\macros\print.wcm")
```

```
// The macro file called print.wcm performs a separate function after the merge.
```

---

## Example for DEFAULTUNITS

APPLICATION (WP;WPWP;DEFAULT)

DEFAULTUNITS (Inches)

GETSTRING (Boxinfo;"Please type text box data";"Text Box")

GETNUMBER (Width;"Enter width of the text box";"WIDTH")

GETNUMBER (Height;"Enter height of the text box";"HEIGHT")

TextBoxCreate ()

BoxPosition (AutoMode:SetBoth!; Width:Width; Height:Height)

Type (Boxinfo)

Close (Yes!)



---

## Example for FOR and ENDFOR

APPLICATION (WP;WPWP;Default)

GETNUMBER (Amount;"How many copies do you want?";"LABELS")

FOR (Count;1;Count<=Amount;Count+1)

Type("Ned Davis")

HardReturn()

Type("1406 West 6th Street")

HardReturn()

Type("Cleveland, OH 44113")

HardPageBreak()

ENDFOR

---

## Example for IF and ENDIF

```
APPLICATION (WP;WPWP;DEFAULT)
```

```
GetWPData(DocName;Name!)
```

```
IF(DocName="")
```

```
PROMPT("Counter Macro";"Your file has not been saved; save before continuing";1;;)
```

```
PAUSE
```

```
ENDPROMPT
```

```
QUIT
```

```
ENDIF
```

---

## Example for PROMPT and ENDPROMPT

```
APPLICATION (WP;WPWP;DEFAULT)
```

```
GetWPData(DocName;Name!)
```

```
IF(DocName="")
```

```
PROMPT("Counter Macro";"Your file has not been saved; save before continuing";1;;)
```

```
PAUSE
```

```
ENDPROMPT
```

```
QUIT
```

```
ENDIF
```

---

## Example for WHILE and ENDWHILE

```
APPLICATION (WP;WPWP;Default)
ASSIGN (Again;"Y")
ASSIGN (Create;"CREATING MERGE SECONDARY FILE")
WHILE ((Again="Y") OR (Again="y"))
  FOREACH (Item;{"Name";"Street";"City";"State";"Zip";"Telephone"})
    GETSTRING (X;"Enter Customer's " + Item + " .";Create)
    Type (X)      MergeEndField ()
  ENDFOR
  MergeEndRecord ()
  GETSTRING (Again;"Create Another Record?";Create)
ENDWHILE
```

---

## Example for FOREACH

```
APPLICATION (WP;WPWP;Default)
ASSIGN (Again;"Y")
ASSIGN (Create;"CREATING MERGE SECONDARY FILE")
WHILE ((Again="Y") OR (Again="y"))
FOREACH (Item;{"Name";"Street";"City";"State";"Zip";"Telephone"})
GETSTRING (X;"Enter Customer's " + Item + " .";Create)
Type (X)
MergeEndField ()
ENDFOR
MergeEndRecord ()
GETSTRING (Again;"Create Another Record?";Create)
ENDWHILE
```

---

## Example for FRACTION

APPLICATION (WP;WPWP;Default)

GETNUMBER (Total;"Type in the Taxable Amount";"Taxable Amount")

INTEGER (Tax1;Total\*0.0625\*100)

ASSIGN (Tax1;Tax1/100)

FRACTION (Tax2;((Total\*0.0625)\*100))

IF (Tax2>0.49)

ASSIGN (Tax1;Tax1+0.01)

ENDIF

NUMSTR (Total;2;Total)

NUMSTR (Tax1;2;Tax1)

Type (Tax1 + " is tax on " + Total)

---

## Example for GETNUMBER

APPLICATION(WP;WPWP;DEFAULT)

GETNUMBER (Copies;"Enter number of copies to print";"Print")

PrintOptions

(Copies:copies;CopiesBy:Printer!;Binding:0;TextQuality:High!;GraphicsQuality:High!)

PrintFull()

---

## Example for GETSTRING

APPLICATION (WP;WPWP;DEFAULT)

GETSTRING (Name;LENGTH=12;"Enter name of file to be saved";"File to be saved")

FileSave (Filename:"c:" +name;ExportType:WordPerfect51!)

FileSave (Filename:"a:" +name;ExportType:WordPerfect51!)



---

## Example for GETUNITS

APPLICATION (WP;WPWP;DEFAULT)

BEEP

DEFAULTUNITS(Inches)

GETUNITS(width;"Enter width of the text box (in inches)";"WIDTH")

BEEP

GETUNITS(height;"Enter height of the text box (in inches)";"HEIGHT")

TextBoxCreate()

BoxPosition (AutoMode:SetBoth!; Width:width; Height:height)

MacroStatusPrompt (On!;"Type Information -- Press Ctrl-F4 to Exit.")

PauseKey (Close!)

Close (Yes!)

---

## Example for GO and LABEL

APPLICATION (WP;WPWP;DEFAULT)

GO (First@)

LABEL (Last@)

Type ("Doe")

QUIT

LABEL (First@)

Type ("John ")

GO (Last@)

---

## Example for INTEGER

APPLICATION (WP;WPWP;Default)

GETNUMBER (Total;"Type in the Taxable Amount";"Taxable Amount")

INTEGER(Tax1;Total\*0.0625\*100)

ASSIGN(Tax1;Tax1/100)

FRACTION(Tax2;((Total\*0.0625)\*100))

IF(Tax2>0.49)

ASSIGN (Tax1;Tax1+0.01)

ENDIF

NUMSTR (Total;2;Total)

NUMSTR (Tax1;2;Tax1)

Type (Tax1 + " is tax on " + Total)

---

## Example for NUMSTR

APPLICATION(WP;WPWP;DEFAULT;)

PosDocBottom()

GetWPData (Lastpage;Page!)

NUMSTR (Lastpage;0;Lastpage)

PosDoctop()

HeaderFooter (Operation:Create!;Item:Footerb!)

PageNumberInsert()

Type(" of " + Lastpage)

Close()

---

## Example for ONCANCEL

```
APPLICATION (WP;WPWP;Default)
ONCANCEL (Name@)
LABEL (DocName@)
GetWPData (Name;Name!)
IF (Name <> "")
PosDocBottom ()
HardReturn ()
HardReturn ()
Type (Name)
ELSE
ASSERTCANCEL
ENDIF
QUIT
LABEL (Name@)
PROMPT ("NO NAME";"This file has no name";3;;)
WAIT (30)
ENDPROMPT
FileSaveAsDlg ()
GO (DocName@)
```

---

## Example for ONCANCEL CALL

```
APPLICATION (WP;WPWP;Default)
ONCANCEL CALL (Name@)
LABEL (DocName@)
GetWPData (Name;Name!)
IF (Name <> "")
PosDocBottom ()
HardReturn ()
HardReturn ()
Type (Name)
ELSE
ASSERTCANCEL
GO (DocName@)
ENDIF
QUIT
LABEL (Name@)
PROMPT ("NO NAME";"This file has no name";3;;)
WAIT (30)
ENDPROMPT
FileSaveAsDlg ()
RETURN
```

---

## Example for ONERROR

APPLICATION (WP;WPWP;DEFAULT)

ONERROR (Invalid@)

LABEL (Top@)

GetString (Name;LENGTH=12;"Enter name of file.;"File to be saved")

FileOpen ("c:\" + name;No!)

QUIT

LABEL (Invalid@)

PROMPT ("ERROR";"You entered a filename that does not exist.";1;;)

WAIT (20)

GO (Top@)

---

## Example for ONERROR CALL

APPLICATION (WP;WPWP;DEFAULT)

ONERROR CALL(Invalid@)

LABEL (Top@)

GETSTRING (Name;LENGTH=12;"Enter name of file. ";"File to be saved")

FileOpen ("a:\" + name;No!)

GO (Top@)

LABEL (Invalid@)

PROMPT ("ERROR";"You entered a filename that does not exist.";1;;)

WAIT (20)

RETURN



---

## Example for ONNOTFOUND

APPLICATION (WP;WPWP;DEFAULT)

ONNOTFOUND (Invalid@)

LABEL (Top@)

GETSTRING (Name;Length=12;"Enter name of file: ";"File to be Retrieved")

FileOpen (Name;No!)

QUIT

LABEL (Invalid@)

PROMPT ("ERROR";"You entered a filename that does not exist.";1;;)

WAIT (20)

ENDPROMPT

GO (Top@)

---

## Example for ONNOTFOUND CALL

```
APPLICATION (WP;WPWP;Default)
ONNOTFOUND CALL(Section2@)
QUIT
LABEL(Section1@)
SearchText(SearchString:"Bill Smith"; SearchDirection:Forward!;SearchScope:Extended!)
PosWordPrevious()
PosWordPrevious()
Type(Text:"Name: ")
PosWordNext()
PosWordNext()
GO(Section1@)
LABEL(Section2@)
// ...More code would be here
RETURN
```

---

## Example for PAUSE

```
APPLICATION (WP;WPWP;DEFAULT)
GetWPData(DocName;Name!)
IF(DocName="")
PROMPT("DOCUMENT STATUS";
"This document has not been saved. Please save file.";1;;)
WAIT (50)
ENDPROMPT
QUIT
ENDIF
```

---

## Example for CASE CALL, RETURN and QUIT

```
APPLICATION(WP;WPWP;Default)
PageMargins(Left:4")
Type(Text:"Thank You,")
HardReturn()
HardReturn()
HardReturn()
HardReturn()
MENU(NAME;DIGIT;;;{"Attorney 1";"Attorney 2";"Attorney 3"})
CASE CALL(NAME;{1;Atty1@;2;Atty2@};Atty3@)
HardReturn()
Type(Text:"Attorney at Law")
PageMargins(Left:1")
QUIT
LABEL(Atty1@)
Type(Text:"Attorney 1")
RETURN
LABEL(Atty2@)
Type(Text:"Attorney 2")
RETURN
LABEL(Atty3@)
Type(Text:"Attorney 3")
RETURN
```

---

## Example for RUN

APPLICATION (WP;WPWP;DEFAULT)

MergeExecute("test.pri";"test.sec")

PosDocVeryTop()

RUN("c:\WPWin\macros\print.wcm")

// The macro file called print.wcm performs a separate function after the merge.

---

## Example for SPEED

APPLICATION (WP;WPWP;Default)

FileNew()

GETNUMBER (Time;

"Type SPEED for slide show. 0-600 (0=Fastest 600=Slowest")

SPEED (Time)

FigureCreate ( )

BoxFigureEdit (Filename:"beach1.wpg")

Close (Save:Yes!)

Display (State:On!)

FOREACH (FN;{"bookworm";"duckling";"jet-2";"tbear";"vacation"})

BoxSelect (Type:Figure!;BoxNumber:"1")

BoxEditor (Type:Figure!)

BoxFigureEdit (Filename:FN + ".wpg")

Close (Save:Yes!)

ENDFOR

Close (No!)

---

## Example for STRLEN

```
ASSIGN (Backward;"")
GETSTRING (Text;"Type the the answer to be typed backwards.;"FORWARD")
STRLEN (Lngth;Text)
FOR (Pos;Lngth;Pos>=1;Pos-1)
SUBSTR (CurrentChar;Pos;1;Text)
ASSIGN (Backward;Backward + CurrentChar)
ENDFOR
PROMPT ("Backward Text";"the reverse of " + Text + " is " + Backward + ".");;)
PAUSE
ENDPROMPT
```

---

## Example for STRNUM

APPLICATION (WP;WPWP;Default)

GETSTRING (Num;"Please type the number you would like to add 25 to: ";"NUMBER")

Type (Num)

PosLineBegin()

SelectMode(State:On!)

PosLineEnd()

GetWPData(Number;SelectedText!)

SelectMode(State:Off!)

STRNUM (VarNum;Number)

ASSIGN (VarNum;VarNum+25)

NUMSTR (Number;3;VarNum)

Type (" + 25 = " + Number)



---

## Example for STRPOS

APPLICATION (WP;WPWP;Default)

GETSTRING (FullName;"Enter First and Last Name.;"FULLNAME")

GETSTRING (Mail;"Enter Mail Stop";"Mail Stop")

Type ("TO: " + FullName)

HardReturn ()

Type ("MAIL STOP: " + Mail)

HardReturn ()

HardReturn ()

STRPOS (Pos;" ";FullName)

SUBSTR (FirstName;1;Pos-1;FullName)

Type ("Dear " + FirstName + " :")

HardReturn ()

HardReturn ()

---

## Example for SUBBYTE

APPLICATION (WP;WPWP;Default)

GETSTRING (Name;"Type your name:","NAME")

SUBBYTE (Name2;1;7;Name)

Type (Name2)

---

## Example for SUBSTR

APPLICATION (WP;WPWP;Default)

GETSTRING (FullName;"Enter First and Last Name.;"FULLNAME")

GETSTRING (Mail;"Enter Mail Stop";"Mail Stop")

Type ("TO: " + FullName)

HardReturn ()

Type ("MAIL STOP: " + Mail)

HardReturn ()

HardReturn ()

STRPOS (Pos;" ";FullName)

SUBSTR (FirstName;1;Pos-1;FullName)

Type ("Dear " + FirstName + ":")

HardReturn ()

HardReturn ()

---

## Example for REPEAT and UNTIL

APPLICATION (WP;WPWP;Default)

Counter:=1

PosDocTop ()

GETNUMBER (Line;"Which line would you like to move to?";"LINE")

REPEAT

PosLineDown ()

Counter:=Counter+1

UNTIL (Counter >= Line)

---

## Example for WAIT

APPLICATION (WP;WPWP;DEFAULT)

ONERROR (Invalid@)

LABEL (Top@)

GetString (Name;LENGTH=12;"Enter name of file";"File to be saved")

FileOpen ("c:\" + name;No!)

QUIT

LABEL (Invalid@)

PROMPT ("ERROR";"You entered a filename that does not exist.";1;;)

WAIT (20)

GO (Top@)

---

## Example for Display

APPLICATION (WP;WPWP;Default)

FileNew()

Display (Off!)

GETNUMBER (Time;"Type SPEED for slide show. 0-600 (0=Fastest 600=Slowest):")

SPEED (Time)

BoxRetrieve (Type:Figure!;Filename:"beach1.wpg")

Display (On!)

FOREACH (FN;{"bookworm";"duckling";"jet-2";"tbear";"vacation"})

BoxSelect (Type:Figure!;BoxNumber:"1")

BoxFigureEdit (Filename:FN + ".wpg")

ENDFOR

Close(No!)

---

## Example for GetWPData

```
APPLICATION(WP;WPWP;DEFAULT;)  
PosDocBottom()  
GetWPData (Lastpage;Page!)  
NUMSTR (Lastpage;0;Lastpage)  
PosDoctop()  
HeaderFooter (Operation:Create!;Item:Footerb!)  
PageNumberInsert()  
Type(" of " + Lastpage)  
Close()
```

---

## Example for MacroStatusPrompt

APPLICATION (WP;WPWP;DEFAULT)

BEEP

DEFAULTUNITS (Inches)

GETUNITS(width;"Enter width of the text box (in inches)";"WIDTH")

BEEP

GETUNITS(height;"Enter height of the text box (in inches)";"HEIGHT")

TextBoxCreate()

BoxPosition (AutoMode:SetBoth!; Width:width; Height:height)

MacroStatusPrompt (On!;"Type Information -- Press Ctrl-F4 to Exit.")

PauseKey (Close!)

Close (Yes!)



---

## Example for MergeVariableGet

APPLICATION (WP;WPWP;DEFAULT)

MergeVariableGet (Amount;"Count")

STRNUM (Amount;Amount)

ASSIGN (Cost;Amount \* 0.29)

NUMSTR (Amount;0;Amount)

NUMSTR (Cost;2;Cost)

PROMPT ("COST";"The number of items is: " + Amount + ". the total cost is: " + Cost;;;)

PAUSE

ENDPROMPT

---

## Example for MergeVariableSet

APPLICATION (WP;WPWP;DEFAULT)

GETSTRING (FN;"Enter file name.")

MergeVariableSet ("FN";FN)

Dialog units are  $\frac{1}{8}$  of system font high and  $\frac{1}{4}$  wide. Consequently, if it takes 20 rows of text in the system font to fill the screen, the screen would be 160 dialog units high.

The part of the document window where you actually work with the text of the document.

A method of displaying an application or document so that many of its elements appear graphically and many features are immediately available as on-screen choices. Different kinds of windows in WordPerfect include document windows, editing windows, and the WordPerfect window. There is only one WordPerfect window and it provides the background for the others. (You can open as many as nine document windows in WordPerfect.)

You may also see the Speller, Thesaurus, and File Manager windows as you use WordPerfect. Each window has a title bar, a menu bar, and a status bar and may contain a scroll bar, Button Bar, and ruler. The Equation Editor window has separate editing and display areas called panes.

To shorten or cut off a character expression that exceeds the specified byte limit.

A feature or option that is turned on or off with the same keystroke or command.

The part of a window or dialog box containing its title.



A display format for open windows. Tiled windows are displayed side-by-side, with no window overlapping any other window.

A file saved in ASCII (DOS) or ANSI (Windows) file format. It contains text, spaces, and returns, but no formatting codes.

In a dialog box, the area in a dialog box where you type text or where WordPerfect types text for you. The selection cursor must be in the text box before you can begin to type.

In a graphics box, a type of graphics box that can be used for placing and rotating text.

The line at the bottom of the WordPerfect window that lists the current font and the position of your insertion point. The status bar also displays some WordPerfect messages and identifies certain features that may be active.

A connection from a spreadsheet file into a WordPerfect document. It lets you transfer information from the spreadsheet into the document. You can update the transferred information as often as necessary to reflect changes made in the spreadsheet file (see DDE link).

A keystroke or keystroke combination that gives you quick access to a WordPerfect feature.

To identify a file, directory, graphics box, or area of text that will be affected by subsequent choices; to identify a dialog box option to be applied to a file, directory graphics box, or area of text. The selected text will change appearance.

The vertical or horizontal bar in a document and in some list boxes that allows you to move rapidly through a document or list by clicking the scroll arrows or dragging the scroll box.



A feature that allows you to display margin markers, tab markers, and buttons for font choices, line spacing, and justification in an area at the top of the window.

A feature that allows you to display text and formatting codes in the document window so that you can edit or check the placement of the codes.

An expression used to evaluate parameters that have only two possible states: TRUE and FALSE.  
A relational expression should not be enclosed in quotation marks.

Radio buttons are like radio push buttons. They appear next to mutually exclusive options, which means that only one of a group can be selected.

A quick list lets you access a directory or group of files without having to enter the full pathname every time you want to locate the same group of files.

A message box containing pertinent information for the user.

A two character expression that specifies which product the subsequent macro commands will utilize.

The software commands that enable an application to communicate with a printer so that the application's information can be printed. WordPerfect for Windows can use WordPerfect printer drivers and Windows printer drivers. WordPerfect printer drivers are identified by a .PRS file extension.



The smallest unit of display on a computer screen.

The location of a certain file or directory in a computer disk drive or on a network. A full pathname includes the drive, root directory, any subdirectory names, and the filename. Each name is separated by a backslash (\). For example, C:\WPWIN\LETTER would refer to the LETTER subdirectory (or file) in the WPWIN directory on the C drive.

The location of a certain file or directory in a computer disk drive or on a network. A full pathname includes the drive, root directory, any subdirectory names, and the filename. Each name is separated by a backslash (\). For example, C:\WPWIN\LETTER would refer to the LETTER subdirectory (or file) in the WPWIN directory on the C drive.

Specifies the characteristics of a command or the effect a command will have. For instance, if you use the `BoxEditor` command to edit a graphics box, you can use a parameter value to specify which type of graphics box to edit.

A section of a window. Panes are used in the Equation Editor, Headers, Footers, Endnotes, Footnotes, Initial Codes, and the Styles Editor.

A group of color selections in a dialog box; or, in the Equation Editor, several groups of commands and symbols that can be inserted into an equation.

A symbol or word that performs a function on one or more expressions. For instance, operators compare expressions, link words together, and perform mathematical functions on numbers.

A system of connecting computers so they can communicate with each other. Files or applications can be stored on one computer, but can be available for access by other computers on the network.



Underlined letters on menu commands or dialog box options which indicate keystroke access for the item or option.

A small box containing a downward-pointing triangle. It is located next to the Maximize button at the right of a window's title bar. Clicking on it will shrink the window to an icon.

A type of dialog box that appears when WordPerfect needs to give information, a warning, or an error message to the user, or to ask for confirmation before continuing to carry out a command.

The area at the top of a window containing the main headings for all the menu items available for that window.

A list of commands that can be applied to the active window or application.

The temporary data storage area for a computer or printer.

A small box at the right of a window title bar. It contains an upward-pointing triangle. When it is clicked, a window that has been at a medium size will expand to fill the screen.

The blinking vertical bar in the active window or dialog box. It indicates where text will appear when you type.



A miniaturized graphic representation of an application or window.

A program item icon is an unopened application in a program group.

An application icon is an application that has been opened and then minimized.

A document icon is a document that has been minimized.

A file that stores setup and startup information for Windows applications.

A set of related options in a dialog box, often with its own subtitle.

The name given to a file of information on disk. The name may be up to eight characters long, with an optional three letter extension, such as MYLETTER.ANN.

The pattern in which a file is organized. Each application creates documents in a unique format which is not accessible by other applications unless those documents are converted to the new application's format.

A document or other type of information that has been saved to a computer disk under an identifying name.

A set of commands used to run a peripheral device such as a printer or monitor.

A type of screen display in which WordPerfect does not show all font attributes and formatting. When Draft Mode is on, you cannot see your document as it will print, but you may be able to work more quickly because the computer does not take as much time updating the appearance of the text.



Disk Operating System (DOS). Software that directs the flow of data between disk drives and your computer. An application needs an operating system to function.

A group of files or directories placed together under one identifying title.

The root directory is the directory on a disk drive to which all other directories are related.

A default directory is the directory where files are saved to or opened and retrieved from if you do not specify a pathname.

A directory tree is a diagram of how directories and subdirectories relate to each other.

Dialog boxes let you communicate with the WordPerfect program. They display warnings and messages and they let you select options and implement those options by choosing appropriate command buttons. Dialog boxes have a title bar and a control menu, but they do not have a menu bar. They can be moved to different locations on the screen. Most dialog boxes must be closed before you can work in the document window, but a few (such as Edit Button Bar) will allow you to move between the window and the dialog box. When a dialog box option is unavailable and cannot be used, it will appear dim.

The screen background and environment for all applications in Windows.

A character that marks the beginning or end of a parameter or character expression.

A Dynamic Data Exchange (DDE) link is a connection between a file created by a Windows application that supports DDE and a WordPerfect document. This connection lets you transfer information from the source file into the WordPerfect document. If both files are open, the transferred information can be updated automatically when information changes in the source file.

Common User Access (CUA) is a set of terminology, keystrokes, and commands designed by IBM to provide standardization among applications. The WordPerfect for Windows default keyboard is CUA compatible.

The menu containing commands that will open, close, maximize, minimize, or restore a window or dialog box.



Formatting commands that make the computer and printer perform various procedures.  
Codes are symbols that the computer and printer understand. They can be edited in Reveal  
Codes.

An area of memory, also called a buffer, where text and commands can be stored to await further action. The contents of the clipboard are erased when you exit Windows.

A check box turns an option on or off. If the check box contains an X, the option is turned on. When text is selected in a document, a shaded check box indicates that the option is on in only part of the selected text. You can select more than one check box in a group.

An additional list of menu items opening from a single menu item.

A WordPerfect feature that enables users to create their own set of buttons and place them in a window for immediate access.

A graphic representation of a command or an option which is activated by clicking on the button with the mouse. If a button is unavailable, it will appear dim.

To transfer macro flow to another part of the macro. Branching is accomplished through the use of programming commands.

A value with only two possible states: TRUE or FALSE.



The end of the currently selected text.

The position a user assigns to a table or graphic.

Document additions that need to be generated, such as cross-references, headers, endnotes, lists, table of contents, and indexes.

An application, window, or dialog box that is currently in use.

Before a macro can play for the first time, it must be compiled. When a macro is compiled, the computer checks all the commands to see if there are any errors and displays an error message if there are. Only a macro with no errors can be played. Some commands are designed to take effect during compilation while others occur during the actual playing of the macro. Each time a macro is edited and saved, it must re-compile before it can be played.

A byte is a storage unit in a computer. Each byte stores the equivalent of one English character; extended characters require more bytes.

The ControlId parameter is used to set the focus (or default, where applicable) on a specific dialog control item. A dialog control will be displayed with the focus if the ControlId parameter of both the control option and the DialogDisplay command are the same .

Valid numbers for the ControlId parameter are 0, 1, 2, and anything greater than 10. The numbers 1 and 2 are used for OK and Cancel push buttons. If 1 or 2 are used, they will turn off the 1 and/or 2 (OK and Cancel) style(s) of the DialogDefine command.

The value of the control (from the ControlId parameter) is assigned to the variable MacroDialogResult. Once the dialog is dismissed, the value of the control used to dismiss the dialog remains in the variable MacroDialogResult. This variable can be tested to see which control dismissed the dialog.

If the ControlId is specified as the number 0 or a string of text, a number greater than 10 is assigned to MacroDialogResult when the **DialogAdd...** command is processed.

## **MacroDialogResult**

Once a dialog box is dismissed, the variable MacroDialogResult will contain a number. The number is determined by the component of the dialog box (the dialog control) that was used to dismiss the box (for example, an OK or a Cancel button). If you do not define the number to be returned when you set the parameters for a control, then the default dialog box numbers will apply. The default dialog box numbers are a 1 if the OK button was pressed; 2 if the Cancel button was pressed. If you wish to test to see what number this variable contains, you must do so before the DialogDestroy command is sent. Once the DialogDestroy command takes effect, the MacroDialogResult variable will be cleared.



### **DialogDefine Example:**

//This will display a basic Dialog with OK and Cancel buttons. It will be placed in the center of the screen and be 200 dialog units wide and 100 high.

```
Application(wp; wppw; default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Test Dialog")
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

## **DialogDisplay Example:**

//This will display a basic Dialog with OK and Cancel buttons. It will be placed in the center of the screen and be 200 dialog\_units wide and 100 high.

```
Application(wp; wppw; default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Test Dialog")
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

### **DialogDestroy Example:**

//This will display a basic Dialog with OK and Cancel buttons. It will be placed in the center of the screen and be 200 dialog\_units wide and 100 high.

```
Application(wp; wppw; default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Test Dialog")
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

## **DialogAddCheckBox Example:**

//Builds a dialog that contains two check boxes. If the user checks the "Option One" box (ControlId 2000), the variable ChkBx1 will be set to 1 and the text in the If statement will be typed.

```
Application(wp; wpwp; default)
```

```
ChkBx1:=0
```

```
ChkBx2:=0
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Check Box Example")
```

```
DialogAddCheckBox(100; 2000; 10; 10; 75; 15; "Option One"; ChkBx1)
```

```
DialogAddCheckBox(100; 2001; 10; 30; 75; 15; "Option Two"; ChkBx2)
```

```
DialogDisplay(100; 2000)
```

```
DialogDestroy(100)
```

```
    If (ChkBx1 = 1)
```

```
        Type("Option One was chosen")
```

```
    EndIf
```

## **DialogAddColorWheel Example:**

//Displays a dialog with a color wheel control. Assigns to variables Blue, Green, and Red their respective color values.

```
Application(wp; wpwp; default)
```

```
Color:=0
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Color Wheel Example")
```

```
DialogAddColorWheel(100; 1001; 10; 10; 75; 50; Color)
```

```
DialogDisplay(100; 1001)
```

```
DialogDestroy(100)
```

```
Blue:=Color/256/256
```

```
Green:=Color/256%256
```

```
Red:=Color%256
```

## **DialogAddComboBox Example:**

//Places two combo boxes in the dialog. the second combo box control is the default when the dialog is displayed and contains three items, the first has three. When the dialog is dismissed, the two items will be typed.

```
Application(wp; wpwp; default)
```

```
Combo1:="Item #1"
```

```
Combo2:=""
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Combo Box Example")
```

```
DialogAddComboBox(100; 2000; 10; 10; 175; 50; 8; Combo1; 10)
```

```
DialogAddListItem(100; 2000; "Item #1")
```

```
DialogAddListItem(100; 2000; "Item #2")
```

```
DialogAddListItem(100; 2000; "Item #3")
```

```
DialogAddComboBox(100; 2001; 10; 30; 175; 50; 16; Combo2; 0)
```

```
DialogAddListItem(100; 2001; "9.5 x 4.0i Envelope")
```

```
DialogAddListItem(100; 2001; "7.5 x 3.88i Monarch")
```

```
DialogDisplay(100; 2001)
```

```
DialogDestroy(100)
```

```
Type(Combo1)
```

```
    HardReturn()
```

```
    Type(Combo2)
```

## **DialogAddCounter Example:**

//Places a counter control in the dialog. The initial value is set to 50 in variable Count. The limits are 0 to 100. The step value when using the increment arrows is 10.

```
Application(wp; wpwp; default)
```

```
Count:=50
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Counter Example")
```

```
DialogAddCounter(100; 1000; 20; 30; 40; 15; 1; Count; 0; 100; 10)
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

```
NumStr(Count; ; Count)
```

```
Type(Count)
```

## **DialogAddEditBox Example:**

//Puts an edit box control in the dialog. sets the WPChars, MultiLine, and WordWrap styles.  
The output in EditReturn is typed.

```
Application(wp; wpwp; default)
```

```
EditReturn:=""
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Edit Box Example")
```

```
DialogAddEditBox(100; 1000; 10; 10; 175; 50; 32+64+1024; EditReturn; 100)
```

```
DialogDisplay(100; 1000)
```

```
DialogDestroy(100)
```

```
Type(EditReturn)
```



## **DialogAddFileNameBox Example:**

//Displays a filename edit control in the dialog. The default filename is "autoexec.bat" as assigned in variable FileNm. The result is typed.

```
Application(wp; wpwp; default)
```

```
FileNm:="autoexec.bat"
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Filename Box Example")
```

```
DialogAddFileNameBox(100; 1000; 20; 20; 125; 15; 0; FileNm; "c:\"; "*.bat")
```

```
DialogDisplay(100; 1000)
```

```
DialogDestroy(100)
```

```
Type(FileNm)
```

### **DialogAddFrame Example:**

//Places a black filled frame in the dialog.

Application(wp; wpwp; default)

DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Frame Control Example")

DialogAddFrame(100; 1000; 10; 10; 175; 50; 4)

DialogDisplay(100; 1000)

DialogDestroy(100)

### **DialogAddGroupBox Example:**

//Displays a group box with the title "Group Box"

Application(wp; wpwp; default)

DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Group Box Example")

DialogAddGroupBox(100; 1000; 10; 10; 175; 45; "Group Box")

DialogDisplay(100; 1000)

DialogDestroy(100)

### **DialogAddHLine Example:**

//Displays a horizontal and vertical line in the dialog.

```
Application(wp;wpwp;default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "HLine and VLine Example")
```

```
DialogAddHLine(100; 1000; 10; 10; 175)
```

```
DialogAddVLine(100; 1001; 100; 15; 50)
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

### **DialogAddVLine Example:**

//Displays a horizontal and vertical line in the dialog.

```
Application(wp;wpwp;default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "HLine and VLine Example")
```

```
DialogAddHLine(100; 1000; 10; 10; 175)
```

```
DialogAddVLine(100; 1001; 100; 15; 50)
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

### **DialogAddHotSpot Example:**

//Places a hot spot in the dialog that will be activated by a single mouse click. If the hot spot is clicked on, the variable Hot will contain the number 1000.

```
Application(wp; wppw; default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Hot Spot Example")
```

```
DialogAddHotSpot(100; 1000; 20; 20; 30; 30; 1)
```

```
DialogDisplay(100; 1000)
```

```
NumStr(Hot; ; MacroDialogResult)
```

```
DialogDestroy(100)
```

```
Type(Hot)
```

### **DialogAddIcon Example:**

//Displays the WP icon in the dialog

Application(wp; wpwp; default)

DllLoad(DllHandle;"c:\WPWin\wpcmcrr.dll")

DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Icon Example")

DialogAddIcon(100; 1000; 22; 22; 30; 30; "WPStamp"; DllHandle)

DialogDisplay(100; 1000)

DialogDestroy(100)

DllFree(DllHandle)

## **DialogAddListBox Example:**

//Places five list items in a list box. Item "Three" is the default item when the dialog is displayed. Neither the Sort or NameSearch styles are used. The list item selected is typed when the dialog is dismissed.

```
Application(wp; wpwp; default)
```

```
List:="Three"
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "List Box Example")
```

```
DialogAddListBox(100; 1000; 10; 10; 175; 55; 0; List)
```

```
DialogAddListItem(100; 1000; "One")
```

```
DialogAddListItem(100; 1000; "Two")
```

```
DialogAddListItem(100; 1000; "Three")
```

```
DialogAddListItem(100; 1000; "Four")
```

```
DialogAddListItem(100; 1000; "Five")
```

```
DialogDisplay(100; 1000)
```

```
DialogDestroy(100)
```

```
Type(List)
```



## **DialogAddListItem Example:**

//Places five list items in a list box. Item "Three" is the default item when the dialog is displayed. Neither the Sort or NameSearch styles are used. The list item selected is typed when the dialog is dismissed.

```
Application(wp; wpwp; default)
```

```
List:="Three"
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "List Box Example")
```

```
DialogAddListBox(100; 1000; 10; 10; 175; 55; 0; List)
```

```
DialogAddListItem(100; 1000; "One")
```

```
DialogAddListItem(100; 1000; "Two")
```

```
DialogAddListItem(100; 1000; "Three")
```

```
DialogAddListItem(100; 1000; "Four")
```

```
DialogAddListItem(100; 1000; "Five")
```

```
DialogDisplay(100; 1000)
```

```
DialogDestroy(100)
```

```
Type(List)
```

## **DialogAddPopUpButton Example:**

//Places five list items in a pop-up list box. Item "Three" is the default item when the dialog is displayed. The list item selected is typed when the dialog is dismissed. Notice the mnemonics as a result of preceding a character with the "&" character.

```
Application(wp; wpwp; default)
```

```
List:="T&hree"
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Pop Up Button Example")
```

```
DialogAddPopUpButton(100; 1000; 10; 10; 40; 15; List)
```

```
DialogAddListItem(100; 1000; "&One")
```

```
DialogAddListItem(100; 1000; "&Two")
```

```
DialogAddListItem(100; 1000; "T&hree")
```

```
DialogAddListItem(100; 1000; "&Four")
```

```
DialogAddListItem(100; 1000; "F&ive")
```

```
DialogDisplay(100; 1000)
```

```
DialogDestroy(100)
```

```
Type(List)
```

## **DialogAddPushButton Example:**

//Displays two push button controls labeled "Button A" and "Button B". When the dialog is dismissed the button used to dismiss the dialog will be typed.

```
Application(wp; wpwp; default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16+8; "Push Button Example")
```

```
DialogAddPushButton(100; 0; 50; 20; 50; 15; 0; "Button &A")
```

```
B1:=MacroDialogResult
```

```
DialogAddPushButton(100; 0; 50; 40; 50; 15; 1; "Button &B")
```

```
B2:=MacroDialogResult
```

```
DialogDisplay(100; B2)
```

```
If (MacroDialogResult = B1)
```

```
Type("Button A")
```

```
Else
```

```
If (MacroDialogResult = B2)
```

```
Type("Button B")
```

```
Else If (MacroDialogResult = 1)
```

```
Type("OK")
```

```
Else If(MacroDialogResult = 2)
```

```
Type("Cancel")
```

```
EndIf EndIf EndIf EndIf
```

```
DialogDestroy(100)
```

## **DialogAddRadioButton Example:**

//Places three radio buttons in the dialog. The variables Button1, Button2 and Button3 will contain either a 1 or 0 when the dialog is dismissed.

```
Application(wp; wpwp; default)
```

```
Button1:=1
```

```
Button2:=0
```

```
Button3:=0
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Radio Button Example")
```

```
DialogAddRadioButton(100; 1000; 10; 10; 50; 15; "Option &1"; Button1)
```

```
DialogAddRadioButton(100; 1001; 10; 30; 50; 15; "Option &2"; Button2)
```

```
DialogAddRadioButton(100; 1002; 10; 50; 50; 15; "Option &3"; Button3)
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

## **DialogAddScrollBar Example:**

//Places a horizontal scroll bar in the dialog. The range is 1 to 4. The scroll box is positioned at the second position. When the dialog is dismissed, the position of the bar is typed.

```
Application(wp; wpwp; default)
```

```
BarPos:=2
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Scroll Bar Example")
```

```
DialogAddScrollBar(100; 1000; 20; 20; 130; 15; 1+16; BarPos; 1; 4)
```

```
DialogDisplay(100;1000)
```

```
DialogDestroy(100)
```

```
NumStr(BarPos; ; BarPos)
```

```
Type(BarPos)
```

## **DialogAddText Example:**

//Displays two text box controls. The first uses the RecessBox style, the second uses ShadowBox.

```
Application(wp; wpwp; default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Text Control Example")
```

```
DialogAddText(100; 1001; 10; 10; 75; 15; 8; "Recessed Box")
```

```
DialogAddText(100; 1002; 10; 30; 75; 15; 16; "Shadow Box")
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

## **DialogAddViewer Example:**

//Displays a view box that shows the "autoexec.bat" file.

```
Application(wp; wpwp; default)
```

```
DialogDefine(100; 50; 50; 200; 100; 1+2+16; "Viewer Example")
```

```
DialogAddViewer(100; 1000; 10; 10; 75; 50; "c:\autoexec.bat")
```

```
DialogDisplay(100; 1)
```

```
DialogDestroy(100)
```

---

## List of Shipping Macros

Following is a list of macros that ship with WordPerfect 5.2 for Windows. Many of the shipping macros make use of a variety of macro commands and are, consequently, good examples of how to integrate multiple commands. To see the shipping macros, simply open the macro file as you would any WordPerfect document. All macro files have a .wcm extension, so to look at the Add macro, for example, you would simply open the file Add.wcm. Each of the following macros except Codes.wcm, Endfoot.wcm, Footend.wcm, and Labels.wcm are also located on the macros keyboard.

**ADD.WCM** - The Add macro can be used to add another font attribute to the current one. For example, in Reveal Codes you can place the insertion point on a Bold code and use this macro to add another attribute such as Underline.

**BARCODE.WCM** - The Barcode macro prints barcodes at any location you specify in the document.

**BULLET.WCM** - The Bullet macro automatically adds bullets to selected text, then indents the insertion point one tab stop to the right.

**BULLETFD.WCM** - The Bulletdf macro is similar to the Bullet macro except it also allows you to change your default bullet character to any character in the WordPerfect character set. This default remains in effect for both Bulletdf.wcm and Bullet.wcm until you change it again.

**CAPITAL.WCM** - The Capital macro capitalizes the first letter of the word your insertion point is on.

**CODES.WCM** - The Codes macro prints text, comments, and codes from the Reveal Codes screen.

**DELETE.WCM** - The Delete macro deletes the line your insertion point is currently on.

**EDIT.WCM** - The Edit macro lets you easily edit codes. For example, in Reveal Codes you can place the insertion point on a figure box code and use this macro to take you quickly to the figure box editor.

**ENDFOOT.WCM** - The Endfoot macro converts endnotes to footnotes. It can convert either the entire document or a selected section.

**ENVELOPE.WCM** - The Envelope macro shipped with WPWin 5.1, but has been enhanced to work with Windows printer drivers. The macro also offers multiple return addresses, multiple envelope sizes, and can re-scan the document if it finds the wrong address the first time. This macro also allows you to append to a document, add a bar code, and print the envelope.

**FILENAME.WCM** - The Filename macro inserts the path and document name at the insertion point.

**FIND.WCM** - The Find macro searches for and deletes a specified bookmark in the current document. See also Mark.wcm.

**FOOTEND.WCM** - The Footend macro converts footnotes to endnotes. It can convert either



the entire document or a selected section.

**GLOSEDIT.WCM** - The Glosedit macro lets you edit glossary entries.

**GLOSSARY.WCM** - The Glossary macro converts user-defined abbreviations to longer text strings, such as converting "WPC" to "WordPerfect Corporation" or "ASAP" into "as soon as possible." You can open this macro to see and/or edit the other abbreviations it expands.

**GOTODOS.WCM** - The Gottodos macro lets you go to DOS from the WPWin document window.

**INLINE.WCM** - The Inline macro sets up the necessary codes for creating an equation in a line of text. It then places you in the Equation Editor.

**INSERT.WCM** - The Insert macro inserts a blank line at the insertion point.

**JUSTIFY.WCM** - The Justify macro opens a menu that lets you choose left, right, or center justification and inserts the code at the insertion point.

**LABELS.WCM** - The Labels macro helps you define a paper size and type for printing labels. It allows you to select your label size from a pre-defined list of common labels for both laser and line printers. It lets you define the orientation and location of the paper and insert the paper size/type code into the document for immediate use.

**LINENUM.WCM** - The Linenum macro asks for a line number, then moves to that position. Linenum.wcm is useful for writing and editing macros because you can note the line number given in a compilation error message, then advance quickly to the location of the error.

**MARK.WCM** - The Mark macro inserts a bookmark into the current document. See also Find.wcm.

**MEMO.WCM** - The Memo macro shipped with WPWin 5.1 but has been enhanced to offer more form types, including a fax cover sheet.

**PAPER.WCM** - The Paper macro opens the Paper Size dialog box.

**PLEADING.WCM** - The Pleading macro creates a style for pleading papers with options such as line numbering and left and right margin lines.

**RECALC.WCM** - The Recalc macro recalculates all the totals in a table if you've changed any of the values.

**REPLACE.WCM** - The Replace macro shipped with WPWin 5.1, but for WPWin 5.2 it has a new interface which makes it easier to search and replace font attributes.

**SMQUOTE.WCM** - The Smquote (Smart Quotes) macro is a combination macro/.DLL that traps the single and double quotation keystrokes and substitutes appropriate left and right typeset-quality characters. The SmartQuotes macro lets the user turn the feature on and off and design how it works.

**TABSET.WCM** - The Tabset macro opens the Tab Set dialog box.

**TRANSPOSE.WCM** - The Transpose macro transposes two characters.

---

## Macro Substitution

The Macro Command Substitution facility allows you to substitute macros for any of the basic functions (such as File Open) used by WordPerfect. The facility is activated by the macro command, "Activate Substitution" which causes WordPerfect to look for substitution entries in the .INI files. It is deactivated with the command, "Deactivate Substitution." When the facility is activated, if it finds a function that is flagged for substitution, it will cause the user-designated macro file to be executed in place of the original function.

Basic functions in WordPerfect are flagged for substitution by placing a section in the WPC.INI file in the following format:

```
[WPWP-Substitute]
(function name)=(macro file name)
```

When you insert the following command into a macro, the substitution facility will read the WPC.INI file, looking for the substitution section described above.

```
UserFunction("WPTP:Activate Substitution")
```

If it finds such a section, it will read the entries and map the functions listed with their corresponding substitution macro file names. After that, whenever any of the substituted functions are used, the macro file will be executed instead.

A substitution can be disabled by using the following command in a macro:

```
UserFunction("WPTP:Deactivate Substitution")
```

UserFunction

ExecuteFunction

ShowMe!

## MacroSubstitution Example::

The File Open and About dialogs will be replaced by simple message boxes in this example. First, you create a substitution section in the WPC.INI file as follows:

```
[WPWP-Substitute]
FileOpenDlg=c:\WPWin\macros\MyFOpen.wcm
AboutDlg=c:\WPWin\macros\MyAbout.wcm
```

Note that the entries are not case sensitive.

Next, you create a macro that will activate the substitution facility and cause the entries to be read, as follows:

```
Application (wp; wpwp; default; "wpwpus.wcd")
UserFunction ("WPTP:Activate Substitution")
```

Then, you create two macros, the MyFOpen and MyAbout macros, each containing a simple message prompt indicating that the command has been intercepted and substituted, such as the following:

```
Application (wp; wpwp; default; "wpwpus.wcd")
PROMPT ("Token Substitution"; "File Open Dialog has been intercepted."; 3;;)
PAUSE
ENDPROMPT
```

By playing the Activate macro, the File Open and About dialogs are substituted by the macro files indicated, and the message prompt appears instead of the dialog boxes.

To deactivate the substitution facility, you create and play a macro such as this one:

```
Application (wp; wpwp; default; "wpwpus.wcd")
UserFunction ("WPTP:Deactivate Substitution")
```

WordPerfect will not remember the substitution from one session to the next. You must play the macro containing the Activate Substitution UserFunction command each time you begin a new session. Multiple calls to the Activate Substitution and Deactivate Substitution user functions are allowed.

