

## Corel VENTURA Commands and Functions

A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z

[ActivateLibrary command](#)

[AppWindowPosition command](#)

[AppWindowPositionGet function](#)

[Bitmap2DBandPass command](#)

[Bitmap2DDisplace command](#)

[Bitmap2DEdgeDetect command](#)

[Bitmap2DEndShear command](#)

[Bitmap2DEndUserDefined command](#)

[Bitmap2DOffset command](#)

[Bitmap2DPixelate command](#)

[Bitmap2DPuzzle command](#)

[Bitmap2DRipple command](#)

[Bitmap2DShear command](#)

[Bitmap2DShearTable command](#)

[Bitmap2DSwirl command](#)

[Bitmap2DTile command](#)

[Bitmap2DTraceContour command](#)

[Bitmap2DUserDefined command](#)

[Bitmap2DUserDefinedPoint command](#)

[Bitmap2DWetPaint command](#)

[Bitmap2DWhirlpool command](#)

[Bitmap2DWind command](#)

[Bitmap3DEmboss command](#)

[Bitmap3DEndMeshWarp command](#)

[Bitmap3DMapToObject command](#)

[Bitmap3DMeshPoint command](#)  
[Bitmap3DMeshWarp command](#)  
[Bitmap3DPageCurl command](#)  
[Bitmap3DPerspective command](#)  
[Bitmap3DPinchPunch command](#)  
[Bitmap3DRotate command](#)  
[Bitmap3DZigZag command](#)  
[BitmapAdjustAutoEqualize command](#)  
[BitmapAdjustBCI command](#)  
[BitmapAdjustColorBalance command](#)  
[BitmapAdjustDesaturate command](#)  
[BitmapAdjustEndEqualize command](#)  
[BitmapAdjustEndHSL command](#)  
[BitmapAdjustEndSampleTargetBalance command](#)  
[BitmapAdjustEndToneCurve command](#)  
[BitmapAdjustEqualize command](#)  
[BitmapAdjustEqualizeChannel command](#)  
[BitmapAdjustGamma command](#)  
[BitmapAdjustHSLChannel command](#)  
[BitmapAdjustHueSaturationLightness command](#)  
[BitmapAdjustReplaceColors command](#)  
[BitmapAdjustSampleTargetBalance command](#)  
[BitmapAdjustSampleTargetColor command](#)  
[BitmapAdjustToneCurve command](#)  
[BitmapAdjustToneTable command](#)  
[BitmapArtisticCanvas command](#)  
[BitmapArtisticGlassBlock command](#)  
[BitmapArtisticImpressionist command](#)  
[BitmapArtisticSmokedGlass command](#)  
[BitmapArtisticVignette command](#)  
[BitmapBlurDirectionalSmooth command](#)  
[BitmapBlurGaussian command](#)  
[BitmapBlurJaggy command](#)  
[BitmapBlurLowPass command](#)  
[BitmapBlurMotion command](#)  
[BitmapBlurRadial command](#)  
[BitmapBlurSmooth command](#)  
[BitmapBlurSoften command](#)  
[BitmapColorTransformBitPlanes command](#)  
[BitmapColorTransformHalftone command](#)  
[BitmapColorTransformPsychedelic command](#)  
[BitmapColorTransformSolarize command](#)  
[BitmapNoiseAdd command](#)  
[BitmapNoiseDiffuse command](#)  
[BitmapNoiseDustScratch command](#)  
[BitmapNoiseMaximum command](#)  
[BitmapNoiseMedian command](#)  
[BitmapNoiseMinimum command](#)  
[BitmapNoiseRemove command](#)  
[BitmapNoiseRemoveMoire command](#)  
[BitmapRender3DStereoNoise command](#)  
[BitmapRenderEndLighting command](#)  
[BitmapRenderLensFlare command](#)  
[BitmapRenderLighting command](#)

[BitmapRenderLightingSource command](#)  
[BitmapSharpen command](#)  
[BitmapSharpenAdaptiveUnsharp command](#)  
[BitmapSharpenDirectionalSharpen command](#)  
[BitmapSharpenFindEdges command](#)  
[BitmapSharpenHighPass command](#)  
[BitmapSharpenUnsharpMask command](#)  
[BitmapTransformDeinterlace command](#)  
[BitmapTransformInvert command](#)  
[BitmapTransformPosterize command](#)  
[BitmapTransformThreshold command](#)  
[BorderTagAddNew command](#)  
[BorderTagCopy command](#)  
[BorderTagCount function](#)  
[BorderTagDelete command](#)  
[BorderTagGetAt function](#)  
[BorderTagRename command](#)

[CAOInstall command and function](#)  
[CAOInvoke command and function](#)  
[CAORegister command and function](#)  
[CAORun command and function](#)  
[CAOUninstall command and function](#)  
[ChapterAdd command](#)  
[ChapterAddNew command](#)  
[ChapterCopy command](#)  
[ChapterCount function](#)  
[ChapterDelete command](#)  
[ChapterGetAt function](#)  
[ChapterPageCount function](#)  
[ChapterPaste command](#)  
[ChapterRename command](#)  
[CharacterTagAddNew command](#)  
[CharacterTagCopy command](#)  
[CharacterTagCount function](#)  
[CharacterTagDelete command](#)  
[CharacterTagGetAt function](#)  
[CharacterTagRename command](#)  
[ClearTransforms command](#)  
[ColorDialog command and function](#)  
[ConvertColorToRGB command](#)  
[ConvertShapeToPicture command](#)  
[CountLibraries function](#)  
[CountWindows function](#)  
[CrossReferenceGet function](#)  
[CurrentChapter function](#)  
[CurrentFrameIndex function](#)  
[CurrentMasterPage function](#)  
[CurrentPageNumber function](#)  
[CurrentPictureFile function](#)  
[CurrentTextFile function](#)  
[CurrentView function](#)  
[CustomAddButton command](#)  
[CustomAddMenu command](#)

[CustomGetCommandAt function](#)  
[CustomGetMenuAt function](#)  
[CustomGetMenuCount function](#)  
[CustomShowToolbar command](#)

[DateTimeGet function](#)  
[DeleteCharacter command](#)  
[DrawArtisticText command](#)  
[DrawCallout command](#)  
[DrawCurveAddNode command](#)  
[DrawCurveBegin command](#)  
[DrawCurveClosePath command](#)  
[DrawCurveCurveTo command](#)  
[DrawCurveEnd command](#)  
[DrawCurveLineTo command](#)  
[DrawCurveMoveTo command](#)  
[DrawEllipse command](#)  
[DrawPolygon command](#)  
[DrawRectangle command](#)  
[DuplicateObject command](#)

[EditAlign command](#)  
[EditCanRedo function](#)  
[EditCanUndo function](#)  
[EditCenter command](#)  
[EditConvertShapeToFrame command](#)  
[EditConvertToCurves command](#)  
[EditConvertToPresetShape command](#)  
[EditConvertVectorToShape command](#)  
[EditCopy command](#)  
[EditCut command](#)  
[EditDelete command](#)  
[EditDuplicate command](#)  
[EditFindTag command and function](#)  
[EditFindText command and function](#)  
[EditGroup command](#)  
[EditInitFindAttribute command](#)  
[EditInitFindColor command](#)  
[EditInitFindTag command](#)  
[EditInitFindTagOptions command](#)  
[EditInitFindText command](#)  
[EditInitReplaceAttribute command](#)  
[EditInitReplaceColor command](#)  
[EditMakeSameSize command](#)  
[EditManageOverrides command](#)  
[EditObject command](#)  
[EditObjectInfo function](#)  
[EditObjectInfoGet command](#)  
[EditOrderBackOne command](#)  
[EditOrderForwardOne command](#)  
[EditOrderReverseOrder command](#)  
[EditOrderToBack command](#)  
[EditOrderToFront command](#)  
[EditPaste command](#)

[EditPasteLink command](#)  
[EditPasteSpecial command](#)  
[EditRedo command](#)  
[EditRepeat command](#)  
[EditReplace command](#)  
[EditSelectAll command](#)  
[EditSelectAllLinked command](#)  
[EditSpace command](#)  
[EditUndo command](#)  
[EditUngroup command](#)  
[EditUpdateNow command](#)

[FileClose command](#)  
[FileExit command](#)  
[FileExport command](#)  
[FileExportPicture command](#)  
[FileImportPicture command](#)  
[FileImportPictureCrop command](#)  
[FileImportPictureResample command](#)  
[FileImportText command](#)  
[FileLibraryClose command](#)  
[FileLibraryCopy command](#)  
[FileLibraryCreateItem command](#)  
[FileLibraryCut command](#)  
[FileLibraryDelete command](#)  
[FileLibraryFindItem command](#)  
[FileLibraryItemCount function](#)  
[FileLibraryItemProperties command](#)  
[FileLibraryItemPropertiesGet function](#)  
[FileLibraryNew command](#)  
[FileLibraryOpen command](#)  
[FileLibraryPaste command](#)  
[FileLibraryPropertiesGet function](#)  
[FileLibrarySave command](#)  
[FileLibrarySelectItem command](#)  
[FileLibraryView command](#)  
[FileLibraryViewFilter command](#)  
[FileLibraryViewSort command](#)  
[FileNew command](#)  
[FileOpen command](#)  
[FilePrint command](#)  
[FilePrintDefault command](#)  
[FilePrintOptions command](#)  
[FilePrintOptionsLayout command](#)  
[FilePrintOptionsPosition command](#)  
[FilePrintOptionsSpecial command](#)  
[FilePrintQuick command](#)  
[FilePublishAs command](#)  
[FilePublishAsAcrobat command](#)  
[FilePublishAsBarista command](#)  
[FilePublishAsCSS command](#)  
[FilePublishAsEnvoy command](#)  
[FilePublishAsHTML command](#)  
[FileRemove command](#)

[FileRename command](#)  
[FileRenameTextFile command](#)  
[FileRevert command](#)  
[FileSave command](#)  
[FileSaveAs command](#)  
[FileSavePageAsEPS command](#)  
[FillFountain command](#)  
[FillFountainColor command](#)  
[FillFountainColorGet function](#)  
[FillFountainColorRemove command](#)  
[FillFountainGet function](#)  
[FillFullColor command](#)  
[FillFullColorGet function](#)  
[FillNoFill command](#)  
[FillOverprint command](#)  
[FillOverprintGet function](#)  
[FillPostscript command](#)  
[FillPostscriptGet function](#)  
[FillTexture command](#)  
[FillTextureGet function](#)  
[FillTwoColor command](#)  
[FillTwoColorGet function](#)  
[FillUniform command](#)  
[FillUniformGet function](#)  
[FormatAttachPicture command](#)  
[FormatAttachTextFile command](#)  
[FormatAutoNumberDefine command](#)  
[FormatAutoNumberGetAt function](#)  
[FormatBitmapBCI command](#)  
[FormatBitmapConvert16Color command](#)  
[FormatBitmapConvert256Color command](#)  
[FormatBitmapConvertBW command](#)  
[FormatBitmapConvertCMYK command](#)  
[FormatBitmapConvertDuotone command](#)  
[FormatBitmapConvertGrayscale command](#)  
[FormatBitmapConvertRGB command](#)  
[FormatBitmapDeskew command](#)  
[FormatBitmapFlip command](#)  
[FormatBitmapHalftone command](#)  
[FormatBitmapResample command](#)  
[FormatBorder command](#)  
[FormatBorderColor command](#)  
[FormatBorderColorGet function](#)  
[FormatBorderGet function](#)  
[FormatBorderOutline command](#)  
[FormatBorderOutlineGet function](#)  
[FormatBorderTagBegin command](#)  
[FormatBorderTagEnd command](#)  
[FormatChapterAutoNumber command](#)  
[FormatChapterAutoNumberGet function](#)  
[FormatChapterCounter command](#)  
[FormatChapterCounterGet function](#)  
[FormatChapterEndnote command](#)  
[FormatChapterEndnoteGet function](#)

[FormatChapterFootnote command](#)  
[FormatChapterFootnoteGet function](#)  
[FormatChapterGeneral command](#)  
[FormatChapterGeneralGet function](#)  
[FormatChapterTypography command](#)  
[FormatChapterTypographyGet function](#)  
[FormatCharTagBegin command](#)  
[FormatCharTagColor command](#)  
[FormatCharTagColorGet function](#)  
[FormatCharTagDefaults command](#)  
[FormatCharTagDefaultsGet function](#)  
[FormatCharTagEnd command](#)  
[FormatCharTagFont command](#)  
[FormatCharTagFontGet function](#)  
[FormatCharTagHyphenation command](#)  
[FormatCharTagHyphenationGet function](#)  
[FormatCondition command](#)  
[FormatConditionGet function](#)  
[FormatCreateFrame command](#)  
[FormatDetachPicture command](#)  
[FormatDetachTextFile command](#)  
[FormatFootnoteDefine command](#)  
[FormatFootnoteSeparator command](#)  
[FormatFootnoteSeparatorGet function](#)  
[FormatFrameAnchor command](#)  
[FormatFrameAnchorGet function](#)  
[FormatFrameColGutterOutline command](#)  
[FormatFrameColGutterOutlineGet function](#)  
[FormatFrameColumnGetAt function](#)  
[FormatFrameColumns command](#)  
[FormatFrameColumnSet command](#)  
[FormatFrameColumnsGet function](#)  
[FormatFrameGeneral command](#)  
[FormatFrameGeneralGet function](#)  
[FormatFrameMarginsInside command](#)  
[FormatFrameMarginsInsideGet function](#)  
[FormatFrameMarginsOutside command](#)  
[FormatFrameMarginsOutsideGet function](#)  
[FormatFramePicture command](#)  
[FormatFramePictureGet function](#)  
[FormatFrameTagBegin command](#)  
[FormatFrameTagEnd command](#)  
[FormatFrameTypography command](#)  
[FormatFrameTypographyGet function](#)  
[FormatFrameVerticalRule command](#)  
[FormatFrameVerticalRuleGet function](#)  
[FormatFrameVerticalRuleGetAt function](#)  
[FormatFrameVerticalRuleSet command](#)  
[FormatFrameVertRuleOutline command](#)  
[FormatFrameVertRuleOutlineGet function](#)  
[FormatGetFrameTag function](#)  
[FormatGetParaTag function](#)  
[FormatIndex command](#)  
[FormatIndexCount function](#)

[FormatIndexDelete command](#)  
[FormatIndexGet function](#)  
[FormatIndexGetAt function](#)  
[FormatIndexRename command](#)  
[FormatIndexUpdate command](#)  
[FormatMasterPage command](#)  
[FormatMasterPageApply command](#)  
[FormatMasterPageGet function](#)  
[FormatMergeTags command](#)  
[FormatObjectBegin command](#)  
[FormatObjectEnd command](#)  
[FormatOverrideCounter command](#)  
[FormatOverrideCounterGet function](#)  
[FormatPageLayout command](#)  
[FormatPageLayoutGet command](#)  
[FormatPageTagBegin command](#)  
[FormatPageTagEnd command](#)  
[FormatParaAlignment command](#)  
[FormatParaAlignmentGet function](#)  
[FormatParaBreaks command](#)  
[FormatParaBreaksGet function](#)  
[FormatParaBullet command](#)  
[FormatParaBulletGet function](#)  
[FormatParaColor command](#)  
[FormatParaColorGet function](#)  
[FormatParaDefaults command](#)  
[FormatParaDefaultsGet function](#)  
[FormatParaDropCap command](#)  
[FormatParaDropCapGet function](#)  
[FormatParaEffectColor command](#)  
[FormatParaEffectColorGet function](#)  
[FormatParaEffectFont command](#)  
[FormatParaEffectFontGet function](#)  
[FormatParaEffects command](#)  
[FormatParaEffectsGet function](#)  
[FormatParaFont command](#)  
[FormatParaFontGet function](#)  
[FormatParaHyphenation command](#)  
[FormatParaHyphenationGet function](#)  
[FormatParaSpacing command](#)  
[FormatParaSpacingGet function](#)  
[FormatParaTab command](#)  
[FormatParaTabClear command](#)  
[FormatParaTabCount function](#)  
[FormatParaTabGet function](#)  
[FormatParaTabGetAt command](#)  
[FormatParaTabSet command](#)  
[FormatParaTagBegin command](#)  
[FormatParaTagEnd command](#)  
[FormatParaTypography command](#)  
[FormatParaTypographyGet function](#)  
[FormatPubCondition command](#)  
[FormatPubConditionAdd command](#)  
[FormatPubConditionDelete command](#)



[FormatPubConditionGetAt function](#)  
[FormatPubConditionRename command](#)  
[FormatPubGeneral command](#)  
[FormatPubGeneralGet function](#)  
[FormatPubText command](#)  
[FormatPubTextAdd command](#)  
[FormatPubTextDelete command](#)  
[FormatPubTextGetAt function](#)  
[FormatPubTextRename command](#)  
[FormatPubVariable command](#)  
[FormatPubVariableAdd command](#)  
[FormatPubVariableDelete command](#)  
[FormatPubVariableGetAt function](#)  
[FormatPubVariableRename command](#)  
[FormatRule command](#)  
[FormatRuleColor command](#)  
[FormatRuleColorGet function](#)  
[FormatRuleGet function](#)  
[FormatRuleOutline command](#)  
[FormatRuleOutlineGet function](#)  
[FormatRuleSide command](#)  
[FormatRuleSideGet function](#)  
[FormatRuleStyle command](#)  
[FormatRuleStyleGet function](#)  
[FormatRuleTag command](#)  
[FormatRuleTagBegin command](#)  
[FormatRuleTagColor command](#)  
[FormatRuleTagColorGet function](#)  
[FormatRuleTagEnd command](#)  
[FormatRuleTagGet function](#)  
[FormatRuleTagOutline command](#)  
[FormatRuleTagOutlineGet function](#)  
[FormatRulingLine command](#)  
[FormatRulingLineColor command](#)  
[FormatRulingLineColorGet function](#)  
[FormatRulingLineGet function](#)  
[FormatRulingLineOutline command](#)  
[FormatRulingLineOutlineGet function](#)  
[FormatRulingLineStyle command](#)  
[FormatRulingLineStyleGet function](#)  
[FormatSelectedText command](#)  
[FormatSelectedTextColor command](#)  
[FormatSelectedTextColorGet function](#)  
[FormatSelectedTextGet function](#)  
[FormatSelectedTextReset command](#)  
[FormatSetCharacterTag command](#)  
[FormatSetCurrentColor command](#)  
[FormatSetFont command](#)  
[FormatSetFrameTag command](#)  
[FormatSetParaTag command](#)  
[FormatTOC command](#)  
[FormatTOCCount function](#)  
[FormatTOCDelete command](#)  
[FormatTOCGet function](#)

[FormatTOCGetAt function](#)  
[FormatTOCLevelGetAt function](#)  
[FormatTOCLevelSet command](#)  
[FormatTOCRename command](#)  
[FormatTOCUpdate command](#)  
[FormatUpdateAutoNumber command](#)  
[FormatUpdatePublication command](#)  
[FrameAutoWrap command](#)  
[FrameContinueTextFlow command](#)  
[FrameCount function](#)  
[FrameFirst command](#)  
[FrameFirstLine command](#)  
[FrameFirstPara command](#)  
[FrameFooter command](#)  
[FrameFootnote command](#)  
[FrameHeader command](#)  
[FrameLastLine command](#)  
[FrameLastPara command](#)  
[FrameMakeSpread command](#)  
[FrameNext command](#)  
[FrameNextLinked command](#)  
[FramePrev command](#)  
[FramePrevLinked command](#)  
[FrameRemoveFromFrame command](#)  
[FrameSelectAll command](#)  
[FrameShowOnFacingPage command](#)  
[FrameSizeToObject command](#)  
[FrameTagAddNew command](#)  
[FrameTagCopy command](#)  
[FrameTagCount function](#)  
[FrameTagDelete command](#)  
[FrameTagGetAt function](#)  
[FrameTagRename command](#)

[GetFillType function](#)  
[GetGraphicType function](#)  
[GetSelectionType function](#)  
[GetTextEntryMode function](#)  
[GraphicCount function](#)  
[GraphicFirst command](#)  
[GraphicNext command](#)  
[GraphicPosition command](#)  
[GraphicPositionGet function](#)  
[GraphicPrev command](#)  
[GraphicRotate command](#)  
[GraphicSkew command](#)

[HiddenTextGet function](#)

[IndexEntryGet function](#)  
[IndexEntryGetAt function](#)  
[InsertCrossReference command](#)  
[InsertDateTime command](#)  
[InsertEndnote](#)

[InsertEquation command](#)  
[InsertFootnote command](#)  
[InsertFrameAnchor command](#)  
[InsertFrameAnchorBegin command](#)  
[InsertHiddenText command](#)  
[InsertHyperlink command](#)  
[InsertIndexAddLevel command](#)  
[InsertIndexBegin command](#)  
[InsertIndexEnd command](#)  
[InsertLineBreak command](#)  
[InsertMarker command](#)  
[InsertObject command](#)  
[InsertParagraph command](#)  
[InsertSymbol command](#)  
[InsertURL command](#)  
[InsertVariable command](#)  
[IsCaretAtEndOfText command](#)  
[IsCaretInTable function](#)  
[IsFormatError function](#)

[LoadStyleSheet command](#)

[MarkerGet function](#)  
[MasterPageAddNew command](#)  
[MasterPageCopy command](#)  
[MasterPageCount function](#)  
[MasterPageDelete command](#)  
[MasterPageGetAt function](#)  
[MasterPageRename command](#)  
[MoveObject command](#)

[NextDocWindow command](#)  
[NudgeObject command](#)

[OleItemCopy command](#)  
[Outline command](#)  
[OutlineColor command](#)  
[OutlineColorGet function](#)  
[OutlineGet function](#)  
[OutlineOverprint command](#)  
[OutlineOverprintGet function](#)

[PageDeletePages command](#)  
[PageEnableFooter command](#)  
[PageEnableFootnotes command](#)  
[PageEnableHeader command](#)  
[PageFacingPageProperties command](#)  
[PageFirstLine command](#)  
[PageFirstPara command](#)  
[PageGrid command](#)  
[PageGridGet function](#)  
[PageGuideline command](#)  
[PageGuidelineAdd command](#)  
[PageGuidelineCount function](#)

[PageGuidelineDelete command](#)  
[PageGuidelineGet function](#)  
[PageGuidelineGetAt function](#)  
[PageInsertPages command](#)  
[PageLastLine command](#)  
[PageLastPara command](#)  
[PageTagAddNew command](#)  
[PageTagApply command](#)  
[PageTagCopy command](#)  
[PageTagCount function](#)  
[PageTagDelete command](#)  
[PageTagGetAt function](#)  
[PageTagRename command](#)  
[ParaTagAddNew command](#)  
[ParaTagCopy command](#)  
[ParaTagCount function](#)  
[ParaTagDelete command](#)  
[ParaTagGetAt function](#)  
[ParaTagRename command](#)  
[PictureFileCopy command](#)  
[PictureFileCount function](#)  
[PictureFileEmbed command](#)  
[PictureFileGetAt function](#)  
[PictureFileIsLinked function](#)  
[PictureFileIsUsed function](#)  
[PrevDocWindow command](#)  
[PubConditionCount function](#)  
[PublicationModified function](#)  
[PublicationName function](#)  
[PubTextCount function](#)  
[PubVariableCount function](#)

[ResizeObject command](#)  
[RotateObject command](#)  
[RuleTagAddNew command](#)  
[RuleTagCopy command](#)  
[RuleTagCount command](#)  
[RuleTagDelete command](#)  
[RuleTagGetAt command](#)  
[RuleTagRename command](#)

[SelectedText function](#)  
[SelectNextObject command](#)  
[SelectObjectAt command](#)  
[SetTextEntryMode command](#)  
[SetVisible command](#)  
[StyleSheetCopy command](#)

[TableAppendColumn command](#)  
[TableAppendRow command](#)  
[TableAutoFill command](#)  
[TableAutoFillSeries command](#)  
[TableAutoFormat command](#)  
[TableAutoSum command](#)

[TableCellBorders command](#)  
[TableColumnResize command](#)  
[TableColumnWidth command](#)  
[TableColumnWidthGet function](#)  
[TableCountTables function](#)  
[TableCreateTable command](#)  
[TableDeleteColumn command](#)  
[TableDeleteRow command](#)  
[TableDeleteTable command](#)  
[TableFirstTable command](#)  
[TableFormatTable command](#)  
[TableFormatTableGet function](#)  
[TableFunction command](#)  
[TableFunctionGet function](#)  
[TableGetCell command](#)  
[TableGetSelection function](#)  
[TableGotoCell command](#)  
[TableInsertColumn command](#)  
[TableInsertRow command](#)  
[TableMergeCells command](#)  
[TableNextTable command](#)  
[TableNormalRule command](#)  
[TablePosition command](#)  
[TablePositionGet function](#)  
[TableRecalculate command](#)  
[TableRemoveColumn command](#)  
[TableRemoveRow command](#)  
[TableSelectCell command](#)  
[TableSelectColumn command](#)  
[TableSelectRow command](#)  
[TableSelectTable command](#)  
[TableSkew command](#)  
[TableSkewGet function](#)  
[TableSkewTopRow command](#)  
[TableSort command](#)  
[TableSpacing command](#)  
[TableSpacingGet function](#)  
[TableSplitCells command](#)  
[TextArrangeSideBySide command](#)  
[TextCapitalize command](#)  
[TextCharDown command](#)  
[TextCharLeft command](#)  
[TextCharRight command](#)  
[TextCharUp command](#)  
[TextEndOfArticle command](#)  
[TextEndOfColumn command](#)  
[TextEndOfFrame command](#)  
[TextEndOfLine command](#)  
[TextEndOfPara command](#)  
[TextExtendSelection command](#)  
[TextFileCopy command](#)  
[TextFileCount function](#)  
[TextFileGetAt function](#)  
[TextFileIsUsed function](#)

[TextFileParaCount function](#)  
[TextFileSelect command](#)  
[TextFileSelectGet function](#)  
[TextIsParaOverriden function](#)  
[TextLineInfoGet function](#)  
[TextLowercase command](#)  
[TextParaDown command](#)  
[TextParaUp command](#)  
[TextSelectParagraph command](#)  
[TextSelectSentence command](#)  
[TextSelectWord command](#)  
[TextSentenceDown command](#)  
[TextSentenceUp command](#)  
[TextSpecialItemType function](#)  
[TextStartOfArticle command](#)  
[TextStartOfColumn command](#)  
[TextStartOfFrame command](#)  
[TextStartOfLine command](#)  
[TextStartOfPara command](#)  
[TextUppercase command](#)  
[TextWordLeft command](#)  
[TextWordRight command](#)  
[ToolsAutoScript command](#)  
[ToolsSpellCheckBegin function](#)  
[ToolsSpellCheckEnd command](#)  
[ToolsSpellSuggestionGetAt function](#)  
[ToolsThesaurusBegin function](#)  
[ToolsThesaurusDefinitionGet function](#)  
[ToolsThesaurusEnd command](#)  
[ToolsThesaurusGetAt function](#)  
[TypeText command](#)

[VariableGet function](#)  
[ViewColorCorrection command](#)  
[ViewCopyEditor command](#)  
[ViewFrameBorders command](#)  
[ViewFreezeScreen command](#)  
[ViewGotoMasterPage command](#)  
[ViewGotoPage command](#)  
[ViewGraphicResolution command](#)  
[ViewHideAllGraphics command](#)  
[ViewMasterPage command](#)  
[ViewPageFooter command](#)  
[ViewPageFootnote command](#)  
[ViewPageHeader command](#)  
[ViewPageLayout command](#)  
[ViewRedraw command](#)  
[ViewRulers command](#)  
[ViewUnFreezeScreen command](#)  
[ViewZoom command](#)  
[ViewZoomLayout command](#)



## Changes from Corel VENTURA 7 to 8

This section of the documentation summarizes changes in Corel VENTURA scripting commands and functions from version 7 to 8:

- [New commands and functions](#)
- [Modified commands and functions](#)
- [Obsolete commands and functions](#)

### New commands and functions

[Bitmap2DEndUserDefined](#)

[Bitmap2DBandPass](#)

[Bitmap2DDisplace](#)

[Bitmap2DEdgeDetect](#)

[Bitmap2DEndShear](#)

[Bitmap2DOffset](#)

[Bitmap2DPixelate](#)

[Bitmap2DPuzzle](#)

[Bitmap2DRipple](#)

[Bitmap2DShearTable](#)

[Bitmap2DShear](#)

[Bitmap2DSwirl](#)

[Bitmap2DTile](#)

[Bitmap2DTraceContour](#)

[Bitmap2DUserDefinedPoint](#)

[Bitmap2DUserDefined](#)

[Bitmap2DWetPaint](#)

[Bitmap2DWhirlpool](#)

[Bitmap2DWind](#)

[Bitmap3DEmboss](#)

[Bitmap3DEndMeshWarp](#)

[Bitmap3DMapToObject](#)

[Bitmap3DMeshPoint](#)

[Bitmap3DMeshWarp](#)

[Bitmap3DPageCurl](#)

[Bitmap3DPerspective](#)

[Bitmap3DPinchPunch](#)

[Bitmap3DRotate](#)

[Bitmap3DZigZag](#)

[BitmapAdjustAutoEqualize](#)

[BitmapAdjustBCI](#)

[BitmapAdjustColorBalance](#)

[BitmapAdjustDesaturate](#)

[BitmapAdjustEndEqualize](#)

[BitmapAdjustEndHSL](#)

[BitmapAdjustEndSampleTargetBalance](#)

[BitmapAdjustEndToneCurve](#)

[BitmapAdjustEqualizeChannel](#)

[BitmapAdjustEqualize](#)

[BitmapAdjustGamma](#)

[BitmapAdjustHSLChannel](#)

[BitmapAdjustHueSaturationLightness](#)

[BitmapAdjustReplaceColors](#)



[BitmapAdjustSampleTargetBalance](#)  
[BitmapAdjustSampleTargetColor](#)  
[BitmapAdjustToneCurve](#)  
[BitmapAdjustToneTable](#)  
[BitmapArtisticCanvas](#)  
[BitmapArtisticGlassBlock](#)  
[BitmapArtisticImpressionist](#)  
[BitmapArtisticSmokedGlass](#)  
[BitmapArtisticVignette](#)  
[BitmapBlurDirectionalSmooth](#)  
[BitmapBlurGaussian](#)  
[BitmapBlurJaggy](#)  
[BitmapBlurLowPass](#)  
[BitmapBlurMotion](#)  
[BitmapBlurRadial](#)  
[BitmapBlurSmooth](#)  
[BitmapBlurSoften](#)  
[BitmapColorTransformBitPlanes](#)  
[BitmapColorTransformHalftone](#)  
[BitmapColorTransformPsychedelic](#)  
[BitmapColorTransformSolarize](#)  
[BitmapNoiseAdd](#)  
[BitmapNoiseDiffuse](#)  
[BitmapNoiseDustScratch](#)  
[BitmapNoiseMaximum](#)  
[BitmapNoiseMedian](#)  
[BitmapNoiseMinimum](#)  
[BitmapNoiseRemoveMoire](#)  
[BitmapNoiseRemove](#)  
[BitmapRender3DStereoNoise](#)  
[BitmapRenderEndLighting](#)  
[BitmapRenderLensFlare](#)  
[BitmapRenderLightingSource](#)  
[BitmapRenderLighting](#)  
[BitmapSharpenAdaptiveUnsharp](#)  
[BitmapSharpenDirectionalSharpen](#)  
[BitmapSharpenFindEdges](#)  
[BitmapSharpenHighPass](#)  
[BitmapSharpenUnsharpMask](#)  
[BitmapSharpen](#)  
[BitmapTransformDeinterlace](#)  
[BitmapTransformInvert](#)  
[BitmapTransformPosterize](#)  
[BitmapTransformThreshold](#)  
[CAOInvoke](#)  
[ChapterPaste](#)  
[ConvertShapeToPicture](#)  
[CrossReferenceGet](#)  
[CurrentFrameIndex](#)  
[DateTimeGet](#)  
[EditInitFindTagOptions](#)  
[FileExportPicture](#)  
[FilePublishAsCSS](#)  
[FileRevert](#)

[FileSavePageAsEPS](#)  
[FormatBitmapBCI](#)  
[FormatBitmapConvert16Color](#)  
[FormatBitmapConvert256Color](#)  
[FormatBitmapConvertBW](#)  
[FormatBitmapConvertCMYK](#)  
[FormatBitmapConvertDuotone](#)  
[FormatBitmapConvertGrayscale](#)  
[FormatBitmapConvertRGB](#)  
[FormatBitmapDeskew](#)  
[FormatBitmapFlip](#)  
[FormatBitmapHalftone](#)  
[FormatBitmapResample](#)  
[FormatObjectBegin](#)  
[FormatObjectEnd](#)  
[FormatPageLayoutGet](#)  
[FormatPageLayout](#)  
[FormatPageTagBegin](#)  
[FormatPageTagEnd](#)  
[FormatRuleColorGet](#)  
[FormatRuleColor](#)  
[FormatRuleGet](#)  
[FormatRuleOutlineGet](#)  
[FormatRuleOutline](#)  
[FormatRuleSideGetF](#)  
[FormatRuleSide](#)  
[FormatRuleStyleGet](#)  
[FormatRuleStyle](#)  
[FormatRuleTagBegin](#)  
[FormatRuleTagColorGet](#)  
[FormatRuleTagColor](#)  
[FormatRuleTagEnd](#)  
[FormatRuleTagGet](#)  
[FormatRuleTagOutlineGet](#)  
[FormatRuleTagOutline](#)  
[FormatRuleTag](#)  
[FormatRule](#)  
[HiddenTextGet](#)  
[IndexEntryGetAt](#)  
[IndexEntryGet](#)  
[InsertHyperlink](#)  
[IsFormatError](#)  
[MarkerGet](#)  
[OleItemCopy](#)  
[PageTagAddNew](#)  
[PageTagApply](#)  
[PageTagCopy](#)  
[PageTagCount](#)  
[PageTagDelete](#)  
[PageTagGetAt](#)  
[PageTagRename](#)  
[PictureFileEmbed](#)  
[PictureFileIsLinked](#)  
[RuleTagAddNew](#)

[RuleTagCopy](#)  
[RuleTagCount](#)  
[RuleTagDelete](#)  
[RuleTagGetAt](#)  
[RuleTagRename](#)  
[VariableGet](#)

#### **Modified commands and functions**

[CurrentPageNumber](#)  
[CurrentPictureFile](#)  
[CurrentTextFile](#)  
[FileExportPicture](#)  
[FileExport](#)  
[FileImportPicture](#)  
[FileImportPicture](#)  
[FileImportText](#)  
[FileImportText](#)  
[FileLibraryItemPropertiesGet](#)  
[FileRemove](#)  
[FileRenameTextFile](#)  
[FileRename](#)  
[FileSaveAs](#)  
[FileSave](#)  
[FormatAttachPicture](#)  
[FormatAttachTextFile](#)  
[FormatCreateFrame](#)  
[FormatDetachPicture](#)  
[FormatDetachTextFile](#)  
[FormatFrameTagBegin](#)  
[FormatParaAlignmentGet](#)  
[FormatParaAlignment](#)  
[FormatParaDropCapGet](#)  
[FormatParaDropCap](#)  
[FormatParaTagBegin](#)  
[FrameContinueTextFlow](#)  
[FrameCount](#)  
[FrameFirst](#)  
[InsertCrossReference](#)  
[PictureFileCopy](#)  
[PictureFileGetAt](#)  
[PictureFileIsLinked](#)  
[PictureFileIsUsed](#)  
[TextFileCopy](#)  
[TextFileGetAt](#)  
[TextFileIsUsed](#)  
[TextFileParaCount](#)  
[TextFileSelectGet](#)  
[TextFileSelect](#)

#### **Obsolete commands and functions**

Many obsolete commands and functions have been replaced by new commands and functions. Please see the obsolete command for its replacement. Some obsolete commands and functions may still work but it is recommended to use the newer commands and functions.

CAORun  
FileImportPictureCrop  
FileImportPictureResample  
FilePublishAsEnvoy  
FilePublishAs  
FormatBorderColorGet  
FormatBorderColor  
FormatBorderGet  
FormatBorderOutlineGet  
FormatBorderOutline  
FormatBorderTagBegin  
FormatBorderTagEnd  
FormatBorder  
FormatFootnoteDefine  
FormatRulingLineColorGet  
FormatRulingLineColor  
FormatRulingLineGet  
FormatRulingLineOutlineGet  
FormatRulingLineOutline  
FormatRulingLineStyleGet  
FormatRulingLineStyle  
FormatRulingLine  
MasterPageAddNew  
MasterPageCopy  
MasterPageCount  
MasterPageDelete  
MasterPageGetAt  
MasterPageRename  
ViewPageFootnote

## **BITMAP MANIPULATING**

## **.FormatBitmapBCI (VENTURA)**

**.FormatBitmapBCI** .Brightness=*long*, .Contrast=*long*, .Intensity=*long*

This command adjusts the brightness, contrast, and intensity of a selected bitmap picture. The command is not available for black & white, 16-color, or 256-color pictures; use Corel PHOTO-PAINT to adjust these types of pictures.

Syntax	Description
.Brightness	Specifies the amount of black included in each color of a picture. Valid values range from -100 to 100 and are specified as the percentage level of contrast in absolute values. If omitted, the default is 0.
.Contrast	Specifies the amount of contrast, which is the visual distance between dark and light tones in an image. Valid values range from -100 to 100 and are specified as the percentage level of contrast in absolute values. If omitted, the default is 0.
.Intensity	Specifies the amount of image intensity. Intensity increases the brightness of lighter pixels and decreases the brightness of mid-tone and dark pixels. Valid values range from -100 to 100 and are specified as the percentage level of intensity in absolute values. If omitted, the default is 0.

### **Note**

- This command corresponds to the Brightness-Contrast-Intensity dialog box in Corel VENTURA. Click Format, Bitmap, Brightness/Contrast.
- Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapBCI 15, 10, -25
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame. The picture then has its B-C-I components adjusted.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Bitmap\_MANIP;;;;";0,"Defaultoverview"),} **Related Topics**

## **.FormatBitmapDeskew (VENTURA)**

### **.FormatBitmapDeskew**

This command adjusts a selected bitmap picture, positioning it squarely on the screen. The **.FormatBitmapDeskew** command is especially useful when working with scanned pictures that, due to imperfect alignment on the scanning surface, appear misaligned.

#### **Note**

- This command corresponds to the Deskew command in the Format menu in Corel VENTURA. Click Format, Bitmap, Deskew.
- Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\Scanned\Scan1.cpt", 0, 0
.FormatBitmapDeskew
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture SCAN1.CPT is imported into the frame and is deskewed.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Bitmap\_MANIP;;;;',0,"Defaultoverview"),} **Related Topics**

## **.FormatBitmapFlip (VENTURA)**

### **.FormatBitmapFlip**

This command flips the bitmap picture horizontally in the selected frame.

#### **Note**

- This command corresponds to the Flip Horizontally command in the Format menu in Corel VENTURA. Click Format, Bitmap, Flip Horizontally.
- Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapFlip
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is flipped horizontally.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Bitmap\_MANIP;;;;',0,"Defaultoverview"),} **Related Topics**



## **.FormatBitmapHalftone (VENTURA)**

**.FormatBitmapHalftone .Cyan=*long*, .Magenta=*long*, .Yellow=*long*, .Radius=*long***

This command converts selected color bitmap pictures to color halftone pictures.

Syntax	Description
<b>.Cyan</b>	Specifies the angle of the cyan color screen. The angle of the screen determines how the color mixes with the other screens. Valid values range from 0 to 359.
<b>.Magenta</b>	Specifies the angle of the magenta color screen. The angle of the screen determines how the color mixes with the other screens. Valid values range from 0 to 359.
<b>.Yellow</b>	Specifies the angle of the yellow color screen. The angle of the screen determines how the color mixes with the other screens. Valid values range from 0 to 359.
<b>.Radius</b>	Specifies the maximum radius of the halftone dot. Valid values range from 2 to 10.

### **Note**

- This command corresponds to the Color Halftone command in the Format menu in Corel VENTURA. Click Format, Bitmap, Color Halftone.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapHalftone 75, 150, 45, 3
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is converted to a halftone.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

**{button ,AL("VENTURA\_Bitmap\_MANIP;;;;','0,"Defaultoverview"),} Related Topics**

## .FormatBitmapResample (VENTURA)

**.FormatBitmapResample .Width=*long*, .Height=*long*, .HorzResolution=*long*, .VertResolution=*long*, .MaintainAspect=*Boolean*, .AntiAlias=*Boolean***

This command adjusts the size and resolution of a selected bitmap picture.

Syntax	Description
<b>.Width</b>	Specifies the width of the bitmap image as a percentage of the original width.
<b>.Height</b>	Specifies the height of the bitmap image as a percentage of the original height.
<b>.HorzResolution</b>	Specifies the horizontal resolution of the imported picture in dots per inch.
<b>.VertResolution</b>	Specifies the vertical resolution of the imported picture in dots per inch.
<b>.MaintainAspect</b>	Specifies whether to maintain the picture's aspect ratio. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0), which is the default if omitted.
<b>.AntiAlias</b>	Specifies the resampling process. Set to TRUE (-1) to enable Anti-alias. Set to FALSE (0) to enable Stretch/Truncate (default if omitted).

### Note



This command corresponds to the Resample dialog box in Corel VENTURA. Click Format, Bitmap, Resample.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapResample 75, 85, 150, 125, FALSE, TRUE
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is resampled.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

**{button ,AL('VENTURA\_Bitmap\_MANIP;;;;','0','Defaultoverview',)} Related Topics**

## **BITMAP CONVERTING**

## **.FormatBitmapConvert16Color (VENTURA)**

### **.FormatBitmapConvert16Color**

This command converts a selected bitmap picture to a 16-color (4-bit) format. Use the 16-color format to create non-photographic images, to print to a low-end color printer, and to maximize your system's memory.

#### **Note**

**I** This command corresponds to the 16-colors command in the Format menu in Corel VENTURA. Click Format, Bitmap, Convert, 16-colors (4-bit).

**I** Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapConvert16Color
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is converted to a 16-color format.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Bitmap\_conv;;;;";0,"Defaultoverview"),} **Related Topics**

## .FormatBitmapConvert256Color (VENTURA)

**.FormatBitmapConvert256Color** .Palette=*long*, .Dither=*long*, .NumOfColor=*long*

This command converts a selected bitmap picture to 256-color. Use 256-colors to create non-photographic images, when printing to a low-end color printer, and to maximize your system's memory.

Syntax	Description
.Palette	Specifies the color palette: 0     Optimized 1     Adaptive 2     Uniform
.Dither	Specifies the dithering type to use in the conversion process: 1     None (default if omitted) 2     Error diffusion 3     Ordered
.NumOfColor	Specifies the number of colors to be included if <b>.Palette</b> is set to Adaptive or Optimized. This parameter is ignored if <b>.Palette</b> is set to Uniform or Custom. The default is 256, if omitted.

### Note

**I** This command corresponds to the Convert to 256 Colors dialog box in Corel VENTURA. Click Format, Bitmap, Convert, 256 colors (8-bit). This command can't be used to create a custom palette.

**I** Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapConvert256Color 1, 2, 102
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is converted to 256-color.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Bitmap\_conv;;;','0,"Defaultoverview"),} Related Topics


## .FormatBitmapConvertBW (VENTURA)


**.FormatBitmapConvertBW .Method=long, .ScreenType=long, .NumOfLines=long, .Angle=long, .Threshold=long**

This command converts a selected bitmap picture to a black & white format.

Syntax	Description
<b>.Method</b>	Specifies the conversion method: 0 Line art 1 Ordered 2 Error diffusion 3 Halftone
<b>.ScreenType</b>	If <b>.Method</b> is set to Halftone, specifies the halftone screen type: 1 Square 2 Round 3 Line 4 Cross 5 Fixed 4x4 (the remaining parameters are ignored if this setting is used) 6 Fixed 8x8 (the remaining parameters are ignored if this setting is used)
<b>.NumOfLines</b>	If <b>.Method</b> is set to Halftone, specifies the number of lines per inch.
<b>.Angle</b>	If <b>.Method</b> is set to Halftone, specifies the halftone screen angle in degrees.
<b>.Threshold</b>	If <b>.Method</b> is set to Line Art, determines the point at which pixels are assigned a white or a black pixel. A grayscale picture is comprised of 256 levels of gray, and each pixel is assigned a grayscale value from 1 (black) to 255 (white). The Threshold option can make more of the pixels white or vice versa. Specify a low value to assign a white pixel in place of all pixels having a grayscale value above that number. Specify a high value to assign a black pixel in place of all pixels having a grayscale value below that number.

### Note

 This command corresponds to the Convert to Black & White dialog box in Corel VENTURA. Click Format, Bitmap, Convert, Black & White (1-bit).

 Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapConvertBW 3, 1, 32, 45
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is converted to a black & white halftone.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Bitmap\_conv;;;;',0,"Defaultoverview"),} **Related Topics**

## **.FormatBitmapConvertCMYK (VENTURA)**

### **.FormatBitmapConvertCMYK**

This command converts a selected bitmap picture to a 32-bit (CMYK) color format. The CMYK model consists of four colors, based on the colors of the inks that are used in four-color printing. Use the CMYK color format to create professional-quality pictures, and when printing to prepress or a CMYK printer.

#### **Note**



This command corresponds to the CMYK Color command in the Format menu in Corel VENTURA. Click Format, Bitmap, Convert, CMYK Color (32-bit).



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapConvertCMYK
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is converted to CMYK.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Bitmap\_conv;;;;',0,"Defaultoverview"),} **Related Topics**

## **.FormatBitmapConvertDuotone (VENTURA)**

### **.FormatBitmapConvertDuotone**

This command converts a selected bitmap picture in grayscale to duotone. The **.FormatBitmapConvertDuotone** command converts pictures using the current settings in the Duotone dialog box only.

#### **Note**

**I** This command corresponds to the Duotone (8-bit) command in the Format menu in Corel VENTURA. Click Format, Bitmap, Convert, Duotone (8-bit).

**I** Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapConvertGrayscale
.FormatBitmapConvertDuotone
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is converted to a grayscale. The picture is then converted to duotone format.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Bitmap\_conv;;;;",0,"Defaultoverview",)} **Related Topics**



## **.FormatBitmapConvertGrayscale (VENTURA)**

### **.FormatBitmapConvertGrayscale**

This command converts a selected bitmap picture to a grayscale format (255 shades of gray), so that it resembles a black and white photograph.

#### **Note**

**I** This command corresponds to the Grayscale command in the Format menu in Corel VENTURA. Click Format, Bitmap, Convert, Grayscale (8-bit).

**I** Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**I** A bitmap picture must be in Grayscale format before it can be converted to a Duotone format. See **.FormatBitmapConvertDuotone** for more information.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapConvertGrayscale
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is converted to Grayscale.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Bitmap\_conv;;;;',0,"Defaultoverview"),} **Related Topics**

## **.FormatBitmapConvertRGB (VENTURA)**

### **.FormatBitmapConvertRGB**

This command converts a selected bitmap picture to a 24-bit (RGB) color format. The RGB color model uses percentages of three colors (red, green, and blue) to create colors. RGB is the most commonly used color mode. Use the RGB color format to create high-quality color pictures and when printing to an RGB or CMY printer.

#### **Note**



This command corresponds to the RGB Color command in the Format menu in Corel VENTURA. Click Format, Bitmap, Convert, RGB Color (24-bit).



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FileImportPicture "C:\MyPics\Canada.cpt", 0, 0
.FormatBitmapConvertRGB
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The Corel PHOTO-PAINT picture CANADA.CPT is imported into the frame and is converted to RGB.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Bitmap\_conv;;;;',0,"Defaultoverview"),} **Related Topics**

## **BITMAP ADJUSTING**



# .BitmapAdjustAutoEqualize (VENTURA)

**.BitmapAdjustAutoEqualize .AutoBlack=*long*, .AutoWhite=*long***

This command equalizes the current image. You can manipulate the tonal range by accentuating or toning down detail in shadow or highlight areas, correcting overexposure or underexposure, or by generally adjusting the tonal range. The Core! VENTURA Auto Equalize command performs a flat equalization by redistributing a significant portion of the tonal range between 0 and 255 automatically.

Syntax	Description
<b>.AutoBlack</b>	Set this parameter to 5.
<b>.AutoWhite</b>	Set this parameter to 5.

**Note**

-  This command was introduced in Core! VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Core! VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.BitmapAdjustAutoEqualize 5, 5`

## **.BitmapAdjustBCI (VENTURA)**

**.BitmapAdjustBCI .Brightness=*long*, .Contrast=*long*, .Intensity=*long***

This command adjusts the brightness, contrast, and intensity of the tones in a bitmap. The Brightness parameter shifts all pixel values up or down the tonal range. When you adjust the brightness, you lighten or darken all colors equally. The Contrast parameter adjusts the distance between your lightest and darkest pixels. When you increase the intensity, you brighten the lighter areas of your bitmap without washing out the dark areas. Contrast and intensity usually go hand-in-hand. An increase in contrast sometimes washes out detail in shadows and highlights, and an increase in intensity can bring it back.

Syntax	Description
<b>.Brightness</b>	Specifies the Brightness of the current image. The value can range from -100 to 100.
<b>.Contrast</b>	Specifies the Contrast of the current image. The value can range from -100 to 100.
<b>.Intensity</b>	Specifies the Intensity of the current image. The value can range from -100 to 100.

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

```
.BitmapAdjustBCI -51, 33, -8
```

---

{button ,AL("VENTURA\_BitmapAdjust;;;;";0,"Defaultoverview"),} [Related Topics](#)

## .BitmapAdjustColorBalance (VENTURA)

**.BitmapAdjustColorBalance .CyanRed=long, .MagentaGreen=long, .YellowBlue=long, .Shadows=Boolean, .Midtones=Boolean, .Highlights=Boolean, .Luminance=Boolean**

This command shifts the colors in a bitmap. This is useful for correcting color casts in your bitmap. For example, if someone's face is too red in your photograph, you could shift values from red to cyan.

Syntax	Description
<b>.CyanRed</b>	Specifies the balance of red in your image. The value can range from -100 to 100.
<b>.MagentaGreen</b>	Specifies the balance of green in your image. The value can range from -100 to 100.
<b>.YellowBlue</b>	Specifies the balance of blue in your image. The value can range from -100 to 100.
<b>.Shadows</b>	Set to TRUE (-1) to apply the color changes to the darkest pixels in the tonal range. Set to FALSE (0) to preserve the color of the dark pixels.
<b>.Midtones</b>	Set to TRUE (-1) to apply the color changes to the midtones of your image. Set to FALSE (0) to preserve the color of the midtone pixels.
<b>.Highlights</b>	Set to TRUE (-1) to apply the color changes to the lightest pixels in the tonal range. Set to FALSE (0) to preserve the color of the light pixels.
<b>.Luminance</b>	Set to TRUE (-1) to maintain the brightness values of your image. If you set this parameter to FALSE (0), the overall lightness or darkness of your image may be affected by color correction.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapAdjustColorBalance -21, 22, 29, TRUE, TRUE, TRUE, TRUE
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.BitmapAdjustDesaturate (VENTURA)**

### **.BitmapAdjustDesaturate**

This command reduces the saturation of each color in your image to 0, which converts each color to its grayscale equivalent. This makes your image appear to be grayscale without having to convert it.

#### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### **Example**

`.BitmapAdjustDesaturate`

---

`{button ,AL('VENTURA_BitmapAdjust;;;;;','0','Defaultoverview'),}` [Related Topics](#)

## .BitmapAdjustEndEqualize (VENTURA)

### .BitmapAdjustEndEqualize

This command equalizes the current image. You can manipulate the tonal range by accentuating or toning down detail in shadow or highlight areas, correcting overexposure or underexposure, or by generally adjusting the tonal range.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustEqualize** command and end with the **BitmapAdjustEndEqualize** command. In between these two commands can be multiple **BitmapAdjustEqualizeChannel** commands.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.BitmapAdjustEqualize 0, 5, 5, TRUE
.BitmapAdjustEqualizeChannel 0, 12, 252, 2, 243, 202
.BitmapAdjustEqualizeChannel 1, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 2, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 3, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 4, 0, 255, 0, 255, 100
.BitmapAdjustEndEqualize
```

---

{button ,AL("VENTURA\_BitmapAdjust;;;;";0,"Defaultoverview"),} [Related Topics](#)



## .BitmapAdjustEndHSL (VENTURA)

### .BitmapAdjustEndHSL

This command adjust the colors in a bitmap using HLS (Hue, Lightness, and Saturation) values.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustHueSaturationLightness** command and end with the **BitmapAdjustEndHSL** command. In between these two commands can be multiple **BitmapAdjustHSLChannel** commands.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.BitmapAdjustHueSaturationLightness 54, -21, 23
.BitmapAdjustHSLChannel 1, 0, 0, 0
.BitmapAdjustHSLChannel 2, 0, 0, 0
.BitmapAdjustHSLChannel 3, 0, 0, 0
.BitmapAdjustHSLChannel 4, 0, 0, 0
.BitmapAdjustHSLChannel 5, -29, -11, 22
.BitmapAdjustHSLChannel 6, 0, 0, 0
.BitmapAdjustHSLChannel 7, 0, 0, 0
.BitmapAdjustEndHSL
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;',0,"Defaultoverview",)} [Related Topics](#)

## .BitmapAdjustEndSampleTargetBalance (VENTURA)

### .BitmapAdjustEndSampleTargetBalance

This command changes a specific color in a bitmap to a target color by shifting the color values in a single channel or in all the channels at once.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustSampleTargetBalance** command and end with the **BitmapAdjustEndSampleTargetBalance** command. In between these two commands can be multiple **BitmapAdjustSampleTargetColor** commands.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.BitmapAdjustSampleTargetBalance 0, FALSE, FALSE, FALSE, FALSE
.BitmapAdjustSampleTargetColor 0, 5, 0, 0, 0, 0
.BitmapAdjustSampleTargetColor 1, 5, 0, 0, 0, 0
.BitmapAdjustSampleTargetColor 2, 5, 127, 127, 127, 0
.BitmapAdjustSampleTargetColor 3, 5, 127, 127, 127, 0
.BitmapAdjustSampleTargetColor 4, 5, 255, 255, 255, 0
.BitmapAdjustSampleTargetColor 5, 5, 255, 255, 255, 0
.BitmapAdjustEndSampleTargetBalance
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;;','0','Defaultoverview'),} [Related Topics](#)

## .BitmapAdjustEndToneCurve (VENTURA)

### .BitmapAdjustEndToneCurve

This command adjusts the tone curve of the current image. The most pronounced changes occur in the midtones. Adjusting the midtones lets you increase the detail in a low-contrast bitmap without affecting the shadows or highlights.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustToneCurve** command and end with the **BitmapAdjustEndToneCurve** command. In between these two commands can be multiple **BitmapAdjustToneTable** commands.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.BitmapAdjustToneCurve
.BitmapAdjustToneTable 0, 87, 87, 87, 87
.BitmapAdjustToneTable 1, 88, 88, 88, 88
.BitmapAdjustToneTable 2, 89, 89, 89, 89
.BitmapAdjustToneTable 3, 90, 90, 90, 90
.BitmapAdjustToneTable 4, 92, 92, 92, 92
... '(more .BitmapAdjustToneTable command)
... '(more .BitmapAdjustToneTable command)
... '(more .BitmapAdjustToneTable command)
.BitmapAdjustToneTable 247, 101, 101, 101, 101
.BitmapAdjustToneTable 248, 99, 99, 99, 99
.BitmapAdjustToneTable 249, 97, 97, 97, 97
.BitmapAdjustToneTable 250, 95, 95, 95, 95
.BitmapAdjustToneTable 251, 93, 93, 93, 93
.BitmapAdjustToneTable 252, 92, 92, 92, 92
.BitmapAdjustToneTable 253, 90, 90, 90, 90
.BitmapAdjustToneTable 254, 89, 89, 89, 89
.BitmapAdjustToneTable 255, 88, 88, 88, 88
.BitmapAdjustEndToneCurve
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;;'0,"Defaultoverview"),} [Related Topics](#)

# .BitmapAdjustEqualize (VENTURA)



**.BitmapAdjustEqualize .Method=long, .AutoBlack=long, .AutoWhite=long, .AutoAdjust=Boolean**

This command equalizes the current image. You can manipulate the tonal range by accentuating or toning down detail in shadow or highlight areas, correcting overexposure or underexposure, or by generally adjusting the tonal range. The Auto Equalize command performs a flat equalization by redistributing a significant portion of the tonal range between 0 and 255 automatically, while the Level Equalization and Tone Curve dialog boxes provide more advanced control.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustEqualize** command and end with the **BitmapAdjustEndEqualize** command. In between these two commands can be multiple **BitmapAdjustEqualizeChannel** commands.

Syntax	Description
<b>.Method</b>	Specifies the equalization method: 0 Proportional 1 Nonproportional
<b>.AutoBlack</b>	Specifies the percentage of outlying pixels at the dark end of the tonal range that Corel VENTURA will ignore when performing an auto-equalization. Valid values range from 0 to 100%.
<b>.AutoWhite</b>	Specifies the percentage of outlying pixels at the light end of the tonal range that Corel VENTURA will ignore when performing an auto-equalization. Valid values range from 0 to 100%.
<b>.AutoAdjust</b>	Set to TRUE (-1) to let Corel PHOTO-PAINT determine the .AutoBlack and .AutoWhite parameters.

## Note

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

## Example

```
.BitmapAdjustEqualize 0, 5, 5, TRUE
.BitmapAdjustEqualizeChannel 0, 12, 252, 2, 243, 202
.BitmapAdjustEqualizeChannel 1, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 2, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 3, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 4, 0, 255, 0, 255, 100
.BitmapAdjustEndEqualize
```

## .BitmapAdjustEqualizeChannel (VENTURA)

**.BitmapAdjustEqualizeChannel .Index=*long*, .InLow=*long*, .InHigh=*long*, .OutLow=*long*, .OutHigh=*long*, .Gamma=*long***

This command equalizes the current image. You can manipulate the tonal range by accentuating or toning down detail in shadow or highlight areas, correcting overexposure or underexposure, or by generally adjusting the tonal range. The Auto Equalize command performs a flat equalization by redistributing a significant portion of the tonal range between 0 and 255 automatically, while the Level Equalization and Tone Curve dialog boxes provide more advanced control.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustEqualize** command and end with the **BitmapAdjustEndEqualize** command. In between these two commands can be multiple **BitmapAdjustEqualizeChannel** commands.

Syntax	Description
<b>.Index</b>	Specifies the equalization channel: 0 = RGB channels 1 = Red channel 2 = Green channel 3 = Blue channel
<b>.InLow</b>	Specifies a clipping range for the darkest pixels in your image. All pixels that fall between this value and the <b>.OutLow</b> value will map to the darkest pixel value. Valid values range from 0 to 254.
<b>.InHigh</b>	Specifies a clipping range for the brightest pixels in your image. All pixels that fall between this value and the <b>.OutHigh</b> value will map to the brightest pixel value. Valid values range from 1 to 255.
<b>.OutLow</b>	Specifies the output brightness value of the darkest pixels in your image. Valid values range from 0 to 254.
<b>.OutHigh</b>	Specifies the output brightness value of the lightest pixels in your image. Valid values range from 1 to 255.
<b>.Gamma</b>	Specifies the gamma curve value. Adjusting the gamma curve value allows you to pick up detail in a low contrast image without significantly affecting the shadows or highlights. It does affect all the values in your image, but is curve-based so the changes are weighted toward the midtones. Valid values range from 1 to 1000.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapAdjustEqualize 0, 5, 5, TRUE
.BitmapAdjustEqualizeChannel 0, 12, 252, 2, 243, 202
.BitmapAdjustEqualizeChannel 1, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 2, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 3, 0, 255, 0, 255, 100
.BitmapAdjustEqualizeChannel 4, 0, 255, 0, 255, 100
.BitmapAdjustEndEqualize
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;',0,"Defaultoverview"),} [Related Topics](#)

## .BitmapAdjustGamma (VENTURA)

.BitmapAdjustGamma .Value=*long*

This command picks out detail in low contrast bitmaps without significantly affecting shadows or highlights. It does affect all the values in your bitmap, but is curve-based so that the changes are weighted toward the midtones.

Syntax	Description
.Value	Specifies the degree of gamma correction. Valid values range from 0.1 to 10 in the dialog box. In a script this value should be multiplied by 10.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapAdjustGamma 250
```

---

{button ,AL("VENTURA\_BitmapAdjust;;;;;0,"Defaultoverview",)} Related Topics

## .BitmapAdjustHSLChannel (VENTURA)

**.BitmapAdjustHSLChannel .Channel=*long*, .Hue=*long*, .Saturation=*long*, .Lightness=*long***

This command adjust the colors in a bitmap using HLS (Hue, Lightness, and Saturation) values.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustHueSaturationLightness** command and end with the **BitmapAdjustEndHSL** command. In between these two commands can be multiple **BitmapAdjustHSLChannel** commands.

Syntax	Description
.Channel	Specifies the channel: 1 red 2 yellow 3 green 4 cyan 5 blue 6 magenta 7 Grayscale
.Hue	Specifies the Hue tolerance. In the HSB color model, hue is the main attribute in a color that distinguishes it from other colors. Blue, green and red, for example, are all hues. Valid values range from -180 to 180.
.Saturation	Specifies the Saturation tolerance. Saturation is the purity of a color. The HSB color model uses Saturation as a component that determines the purity or intensity of a color. The more colors used to mix a color, the duller the color looks. Valid values range from -100 to 100.
.Lightness	Specifies the Lightness tolerance. Lightness is the amount of black or white in a color. Valid values range from -100 to 100.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapAdjustHueSaturationLightness 54, -21, 23
.BitmapAdjustHSLChannel 1, 0, 0, 0
.BitmapAdjustHSLChannel 2, 0, 0, 0
.BitmapAdjustHSLChannel 3, 0, 0, 0
.BitmapAdjustHSLChannel 4, 0, 0, 0
.BitmapAdjustHSLChannel 5, -29, -11, 22
.BitmapAdjustHSLChannel 6, 0, 0, 0
.BitmapAdjustHSLChannel 7, 0, 0, 0
.BitmapAdjustEndHSL
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;',0,"Defaultoverview",)} [Related Topics](#)

## .BitmapAdjustHueSaturationLightness (VENTURA)

**.BitmapAdjustHueSaturationLightness .Hue=*long*, .Saturation=*long*, .Lightness=*long***

This command adjust the colors in a bitmap using HLS (Hue, Lightness, and Saturation) values.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustHueSaturationLightness** command and end with the **BitmapAdjustEndHSL** command. In between these two commands can be multiple **BitmapAdjustHSLChannel** commands.

Syntax	Description
<b>.Hue</b>	Specifies the Hue tolerance. In the HSB color model, hue is the main attribute in a color that distinguishes it from other colors. Blue, green and red, for example, are all hues. Valid values range from -100 to 100.
<b>.Saturation</b>	Specifies the Saturation tolerance. Saturation is the purity of a color. The HSB color model uses Saturation as a component that determines the purity or intensity of a color. The more colors used to mix a color, the duller the color looks. Valid values range from -100 to 100.
<b>.Lightness</b>	Specifies the Lightness tolerance. Lightness is the amount of black or white in a color. Valid values range from -100 to 100.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapAdjustHueSaturationLightness 54, -21, 23
.BitmapAdjustHSLChannel 1, 0, 0, 0
.BitmapAdjustHSLChannel 2, 0, 0, 0
.BitmapAdjustHSLChannel 3, 0, 0, 0
.BitmapAdjustHSLChannel 4, 0, 0, 0
.BitmapAdjustHSLChannel 5, -29, -11, 22
.BitmapAdjustHSLChannel 6, 0, 0, 0
.BitmapAdjustHSLChannel 7, 0, 0, 0
.BitmapAdjustEndHSL
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;;','0','Defaultoverview",,)} [Related Topics](#)



## .BitmapAdjustReplaceColors (VENTURA)

**.BitmapAdjustReplaceColors .OldColorModel=long, .OldColor1=long, .OldColor2=long, .OldColor3=long, .OldColor4=long, .NewColorModel=long, .NewColor1=long, .NewColor2=long, .NewColor3=long, .NewColor4=long, .Range=long, .IgnoreGrayScale=Boolean, .SingleDestColor=Boolean**

This command replaces one color in your image with another color. Depending on the **.Range** value, the command will replace a single color, or shift the entire image from one range of color to another.

Syntax	Description
<b>.OldColorModel</b>	Specifies the current color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.OldColor1</b>	Specifies the first color component for the <b>.OldColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.OldColor2</b>	Specifies the second color component for the <b>.OldColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.OldColor3</b>	Specifies the third color component for the <b>.OldColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.OldColor4</b>	Specifies the fourth color component for the <b>.OldColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.OldColorModel</b>	Specifies the current color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.NewColor1</b>	Specifies the first color component for the <b>.NewColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.NewColor2</b>	Specifies the second color component for the <b>.NewColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.NewColor3</b>	Specifies the third color component for the <b>.NewColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.NewColor4</b>	Specifies the fourth color component for the <b>.NewColorModel</b> parameter. For example, Black

is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

**.Range**

The range of color that will be replaced by the command. Valid values range from 1 to 100.

**.IgnoreGrayScale**

Specifies whether to ignore grayscale. Set to TRUE (-1) to enable this option.

**.SingleDestColor**

Specifies whether to use single destination color. Set to TRUE (-1) to enable this option.

**Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

```
.BitmapAdjustReplaceColors 5, 212, 74, 104, 0, 5, 254, 248, 0, 0, 50, TRUE, FALSE
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;;','0','Defaultoverview'),} [Related Topics](#)

## .BitmapAdjustSampleTargetBalance (VENTURA)

**.BitmapAdjustSampleTargetBalance .Channel=*long*, .UseLow=*Boolean*, .UseMid=*Boolean*, .UseHigh=*Boolean*, .UseAll=*Boolean***

This command changes a specific color in a bitmap to a target color by shifting the color values in a single channel or in all the channels at once.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustSampleTargetBalance** command and end with the **BitmapAdjustEndSampleTargetBalance** command. In between these two commands can be multiple **BitmapAdjustSampleTargetColor** commands.

Syntax	Description
.Channel	Specifies the channel you want to change: 0 All three channels (RGB) at once 1 Red channel 2 Green channel 3 Blue Channel
.UseLow	Set to TRUE (-1) to use the low point color values set with the ImageSTColor commands in the command block.
.UseMid	Set to TRUE (-1) to use the mid point color values set with the ImageSTColor commands in the command block.
.UseHigh	Set to TRUE (-1) to use the high point color values set with the ImageSTColor commands in the command block.
.UseAll	Set to TRUE (-1) to use all of the color values set with the ImageSTColor commands in the command block.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapAdjustSampleTargetBalance 0, FALSE, FALSE, FALSE, FALSE
.BitmapAdjustSampleTargetColor 0, 5, 0, 0, 0, 0
.BitmapAdjustSampleTargetColor 1, 5, 0, 0, 0, 0
.BitmapAdjustSampleTargetColor 2, 5, 127, 127, 127, 0
.BitmapAdjustSampleTargetColor 3, 5, 127, 127, 127, 0
.BitmapAdjustSampleTargetColor 4, 5, 255, 255, 255, 0
.BitmapAdjustSampleTargetColor 5, 5, 255, 255, 255, 0
.BitmapAdjustEndSampleTargetBalance
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;;',0,"Defaultoverview"),} [Related Topics](#)

## .BitmapAdjustSampleTargetColor (VENTURA)

**.BitmapAdjustSampleTargetColor .Index=*long*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long***

This command changes a specific color in a bitmap to a target color by shifting the color values in a single channel or in all the channels at once.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustSampleTargetBalance** command and end with the **BitmapAdjustEndSampleTargetBalance** command. In between these two commands can be multiple **BitmapAdjustSampleTargetColor** commands.

Syntax	Description
.Index	Specifies the source or target colors to define: 0 Source low color 1 Target low color 2 Source mid color 3 Target mid color 4 Source high color 5 Target high color
.ColorModel	Specifies the color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapAdjustSampleTargetBalance 0, FALSE, FALSE, FALSE, FALSE
.BitmapAdjustSampleTargetColor 0, 5, 0, 0, 0, 0
.BitmapAdjustSampleTargetColor 1, 5, 0, 0, 0, 0
.BitmapAdjustSampleTargetColor 2, 5, 127, 127, 127, 0
.BitmapAdjustSampleTargetColor 3, 5, 127, 127, 127, 0
.BitmapAdjustSampleTargetColor 4, 5, 255, 255, 255, 0
.BitmapAdjustSampleTargetColor 5, 5, 255, 255, 255, 0
.BitmapAdjustEndSampleTargetBalance
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;',0,"Defaultoverview"),} [Related Topics](#)

## .BitmapAdjustToneCurve (VENTURA)

### .BitmapAdjustToneCurve

This command adjusts the tone curve of the current image. The most pronounced changes occur in the midtones. Adjusting the midtones lets you increase the detail in a low-contrast bitmap without affecting the shadows or highlights.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustToneCurve** command and end with the **BitmapAdjustEndToneCurve** command. In between these two commands can be multiple **BitmapAdjustToneTable** commands.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.BitmapAdjustToneCurve
.BitmapAdjustToneTable 0, 87, 87, 87, 87
.BitmapAdjustToneTable 1, 88, 88, 88, 88
.BitmapAdjustToneTable 2, 89, 89, 89, 89
.BitmapAdjustToneTable 3, 90, 90, 90, 90
.BitmapAdjustToneTable 4, 92, 92, 92, 92
... '(more .BitmapAdjustToneTable command)
... '(more .BitmapAdjustToneTable command)
... '(more .BitmapAdjustToneTable command)
.BitmapAdjustToneTable 247, 101, 101, 101, 101
.BitmapAdjustToneTable 248, 99, 99, 99, 99
.BitmapAdjustToneTable 249, 97, 97, 97, 97
.BitmapAdjustToneTable 250, 95, 95, 95, 95
.BitmapAdjustToneTable 251, 93, 93, 93, 93
.BitmapAdjustToneTable 252, 92, 92, 92, 92
.BitmapAdjustToneTable 253, 90, 90, 90, 90
.BitmapAdjustToneTable 254, 89, 89, 89, 89
.BitmapAdjustToneTable 255, 88, 88, 88, 88
.BitmapAdjustEndToneCurve
```

---

{button ,AL('VENTURA\_BitmapAdjust;;;;;'0,"Defaultoverview"),} [Related Topics](#)

# .BitmapAdjustToneTable (VENTURA)



**.BitmapAdjustToneTable .Number=long, .Curve1=long, .Curve2=long, .Curve3=long, .Curve4=long**

This command adjusts the tone curve of the current image. The most pronounced changes occur in the midtones. Adjusting the midtones lets you increase the detail in a low-contrast bitmap without affecting the shadows or highlights.

This adjustment command is part of a block of commands. The block must begin with the **BitmapAdjustToneCurve** command and end with the **BitmapAdjustEndToneCurve** command. In between these two commands can be multiple **BitmapAdjustToneTable** commands.

Syntax	Description
<b>.Number</b>	Specifies the index number of the individual tone to change.
<b>.Curve1</b>	Specifies the first color component of the tone using the current color model. For example, Red is the first color component for RGB.
<b>.Curve2</b>	Specifies the first color component of the tone using the current color model. For example, Green is the second color component for RGB. If this parameter is not available in the current color model, set it to 0.
<b>.Curve3</b>	Specifies the first color component of the tone using the current color model. For example, Blue is the third color component for RGB. If this parameter is not available in the current color model, set it to 0.
<b>.Curve4</b>	Specifies the first color component of the tone using the current color model. For example, Black is the fourth color component for CMYK. If this parameter is not available in the current color model, set it to 0.

### Note

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapAdjustToneCurve
  .BitmapAdjustToneTable 0, 87, 87, 87, 87
  .BitmapAdjustToneTable 1, 88, 88, 88, 88
  .BitmapAdjustToneTable 2, 89, 89, 89, 89
  .BitmapAdjustToneTable 3, 90, 90, 90, 90
  .BitmapAdjustToneTable 4, 92, 92, 92, 92
  ... '(more .BitmapAdjustToneTable command)
  ... '(more .BitmapAdjustToneTable command)
  ... '(more .BitmapAdjustToneTable command)
  .BitmapAdjustToneTable 247, 101, 101, 101, 101
  .BitmapAdjustToneTable 248, 99, 99, 99, 99
  .BitmapAdjustToneTable 249, 97, 97, 97, 97
  .BitmapAdjustToneTable 250, 95, 95, 95, 95
  .BitmapAdjustToneTable 251, 93, 93, 93, 93
  .BitmapAdjustToneTable 252, 92, 92, 92, 92
  .BitmapAdjustToneTable 253, 90, 90, 90, 90
  .BitmapAdjustToneTable 254, 89, 89, 89, 89
  .BitmapAdjustToneTable 255, 88, 88, 88, 88
.BitmapAdjustEndToneCurve
```

## **BITMAP TRANSFORMING**

## .BitmapTransformDeinterlace (VENTURA)

**.BitmapTransformDeinterlace .ReplaceMode=*long***

This command removes even or odd horizontal lines from scanned or interlaced video bitmaps. You can fill the spaces left by the discarded lines using either of two methods: duplication fills in the spaces with copies of the adjacent lines of pixels, while interpolation fills them in with colors created by averaging the surrounding pixels.

Syntax	Description
<b>.ReplaceMode</b>	Specifies the deinterlace fill method: 0 Replaces even lines using the duplication method 1 Replaces even lines using the interpolation method 2 Replaces odd lines using the duplication method 3 Replaces odd lines using the interpolation method

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapTransformDeinterlace 3
```

---

{button ,AL('VENTURA\_BitmapTransform;;;;','0,"Defaultoverview",)} [Related Topics](#)



## **.BitmapTransformInvert (VENTURA)**

### **.BitmapTransformInvert**

This command inverts the colors in a bitmap, producing an effect much like a photographic negative.

#### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### **Example**

```
.BitmapTransformInvert
```

---

{button ,AL("VENTURA\_BitmapTransform;;;;;0,"Defaultoverview"),} [Related Topics](#)

## .BitmapTransformPosterize (VENTURA)

**.BitmapTransformPosterize .Level=*long***

This command transforms the color range of a bitmap to solid blocks of color, reducing gradual blends to hard edges between areas of color.

Syntax	Description
<b>.Level</b>	Specifies the intensity of the posterization effect (that is, the number of colors in the final image). Valid values range from 2 to 32.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapTransformPosterize 14
```

---

{button ,AL("VENTURA\_BitmapTransform;;;;','0,"Defaultoverview",)} Related Topics

## .BitmapTransformThreshold (VENTURA)

**.BitmapTransformThreshold .Channel=*long*, .LowLevel=*long*, .Threshold=*long*, .HighLevel=*long*, .BiLevel=*long***

This command converts certain shades of each color in an bitmap to black or white. In bi-level mode, the command can convert shades to both black and white at the same time.

Syntax	Description
.Channel	Specifies the channel to convert: 0 All three channels (RGB) at once 1 Red channel 2 Green channel 3 Blue Channel
.Low	Specifies the low-level value. Valid values range from 0 to 255.
.Threshold	Specifies the threshold value. Valid values range from 0 to 255.
.High	Specifies the high-level value. Valid values range from 0 to 255.
.BiLevel	Specifies the conversion method: 0 To black 1 To White 2 Bi-level

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapTransformThreshold 0, 0, 127, 255, 2
```

---

{button ,AL('VENTURA\_BitmapTransform;;;;','0,"Defaultoverview",,)} [Related Topics](#)

## **SPECIAL EFFECTS**

## **SPECIAL EFFECTS 2D EFFECT**

## .Bitmap2DBandPass (VENTURA)

**.Bitmap2DBandPass .InRadius=*long*, .OutRadius=*long*, .InBand=*long*, .MidBand=*long*, .OutBand=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Band Pass filter to adjust the balance of sharp and smooth areas in a bitmap.

Syntax	Description
<b>.InRadius</b>	Specifies the size of the inner band radius. The values range from 1 to 256.
<b>.OutRadius</b>	Specifies the size of the outer band radius. The values range from 1 to 256.
<b>.InBand</b>	Specifies the weighting of the inner band. To eliminate the sharp or smooth areas within a band, set the weighting to 0. Experiment with different weightings to see which provide the best results. The values range from 0 to 100%.
<b>.MidBand</b>	Specifies the weighting of the middle band. The values range from 0 to 100 (%).
<b>.OutBand</b>	Specifies the weighting of the outer band. The values range from 0 to 100 (%).

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.Bitmap2DBandPass 33, 63, 93, 61, 97`

---

{button ,AL('VENTURA\_Bitmap2D;;;;',0,"Defaultoverview"),} [Related Topics](#)

## .Bitmap2DDisplace (VENTURA)

**.Bitmap2DDisplace .FileName=*string*, .Displacement=*long*, .Edges=*long*, .Horizontal=*long*, .Vertical=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Displace filter to alter a bitmap according to a displacement map you choose. Experiment with the settings to create various effects.

Syntax	Description
<b>.Filename</b>	Specifies the name and path of the image file to use as a displacement map.
<b>.Displacement</b>	Specifies the scaling mode of the displacement: 0 Tiles the displacement map over the image 1 Stretches the displacement map to cover the entire image
<b>.Edges</b>	Specifies the method of filling empty areas created by the displacement: 0 Stretches the edge areas 1 Wraps the opposite edge around to the empty areas
<b>.Horizontal</b>	Specifies the horizontal shift. Valid values range from 0 to 100.
<b>.Vertical</b>	Specifies the vertical shift. Valid values range from 0 to 100.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap2DDisplace "c:\custom\displace\worms.pcx", 0, 1, 29, 15
```

---

{button ,AL('VENTURA\_Bitmap2D;;;;',0,"Defaultoverview"),} [Related Topics](#)

## .Bitmap2DEdgeDetect (VENTURA)

**.Bitmap2DEdgeDetect .BackgroundColor=long, .Sensitivity=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long**

This command applies a bitmap effect to a selected bitmap. The Edge Detect filter finds the edges of elements in your bitmap, then converts the edges to lines on a background of a single color. You can use the Edge Detect filter on high-contrast bitmaps, for example, bitmaps that contain text, for the clearest results.

Syntax	Description
<b>.BackgroundColor</b>	Specifies the background color: 0 White 1 Black 2 Other color (when enabled, use the color parameters which follow)
<b>.Sensitivity</b>	Specifies the intensity of the effect. Valid values range from 1 to 10.
<b>.ColorModel</b>	Specifies the color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.Bitmap2DEdgeDetect 2, 2, 5, 0, 146, 64, 0`

---

{button ,AL("VENTURA\_Bitmap2D;;;;",0,"Defaultoverview",)} [Related Topics](#)



## .Bitmap2DEndShear (VENTURA)

### .Bitmap2DEndShear

This command applies a bitmap effect to a selected bitmap. You can use the Shear filter to distort a bitmap along a path that can be customized, or you can use one of the preset shear curves. You can also use customized Shear filters.

The User Defined effect requires a block of commands. The block must begin with the **Bitmap2DShear** command and end with the **Bitmap2DEndShear** command. In between these two commands can be multiple **Bitmap2DShearTable** commands.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.Bitmap2DShear 50, 2, 1, 5, 220, 43, 25, 0
.Bitmap2DShearTable 0, 512
.Bitmap2DShearTable 1, 515
.Bitmap2DShearTable 2, 517
.Bitmap2DShearTable 3, 520
.Bitmap2DShearTable 4, 523
... '(more .Bitmap2DShearTable command)
... '(more .Bitmap2DShearTable command)
... '(more .Bitmap2DShearTable command)
.Bitmap2DShearTable 1019, 501
.Bitmap2DShearTable 1020, 504
.Bitmap2DShearTable 1021, 507
.Bitmap2DShearTable 1022, 509
.Bitmap2DShearTable 1023, 512
.Bitmap2DEndShear
```

---

{button ,AL('VENTURA\_Bitmap2D;;;;',0,"Defaultoverview",)} [Related Topics](#)

## .Bitmap2DEndUserDefined (VENTURA)

### .Bitmap2DEndUserDefined

This command applies a bitmap effect to a selected bitmap. The User Defined filter lets you create your own blur, sharpen, and edge detect special effects, based on values you enter into a 5 X 5 matrix.

The User Defined effect requires a block of commands. The block must begin with the **Bitmap2DUserDefined** command and end with the **Bitmap2DEndUserDefined** command. In between these two commands can be multiple **Bitmap2DUserDefinedPoint** commands.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.Bitmap2DUserDefined 6, 0
.Bitmap2DUserDefinedPoint 0, 0
.Bitmap2DUserDefinedPoint 1, 0
.Bitmap2DUserDefinedPoint 2, 0
.Bitmap2DUserDefinedPoint 3, 0
.Bitmap2DUserDefinedPoint 4, 0
.Bitmap2DUserDefinedPoint 5, 0
.Bitmap2DUserDefinedPoint 6, 3
.Bitmap2DUserDefinedPoint 7, -2
.Bitmap2DUserDefinedPoint 8, 3
.Bitmap2DUserDefinedPoint 9, 0
.Bitmap2DUserDefinedPoint 10, 0
.Bitmap2DUserDefinedPoint 11, -2
.Bitmap2DUserDefinedPoint 12, 2
.Bitmap2DUserDefinedPoint 13, -2
.Bitmap2DUserDefinedPoint 14, 0
.Bitmap2DUserDefinedPoint 15, 0
.Bitmap2DUserDefinedPoint 16, 3
.Bitmap2DUserDefinedPoint 17, -2
.Bitmap2DUserDefinedPoint 18, 3
.Bitmap2DUserDefinedPoint 19, 0
.Bitmap2DUserDefinedPoint 20, 0
.Bitmap2DUserDefinedPoint 21, 0
.Bitmap2DUserDefinedPoint 22, 0
.Bitmap2DUserDefinedPoint 23, 0
.Bitmap2DUserDefinedPoint 24, 0
.Bitmap2DEndUserDefined
```

---

{button ,AL('VENTURA\_Bitmap2D;;;;',0,"Defaultoverview",)} [Related Topics](#)

## .Bitmap2DOffset (VENTURA)

**.Bitmap2DOffset .Horizontal=long, .Vertical=long, .Shift=Boolean, .Edges=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long**

This command applies a bitmap effect to a selected bitmap. You can use the Offset filter to correct bitmap positioning. It shifts the bitmap according to the values set using the Horizontal and Vertical Shift sliders.

Syntax	Description
<b>.Horizontal</b>	Specifies the amount of horizontal shifting. Valid values range from -100 to 100.
<b>.Vertical</b>	Specifies the amount of vertical shifting. Valid values range from -100 to 100.
<b>.Shift</b>	Set to TRUE (-1) to coordinate the horizontal and vertical shift values with the size of the object. With a vertical shift value of 50, the image will shift along the vertical plane a distance corresponding to exactly one-half the size of the image. Set to FALSE (0) for absolute offsetting.
<b>.Edges</b>	Specifies the method used to fill the empty area left behind by the offset: 0 Wraps the opposite edge around to the empty area 1 Stretches the edges of the image to fill the empty area 2 Uses the background paper color
<b>.ColorModel</b>	Specifies the color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.Bitmap2DOffset 67, 30, TRUE, 2, 5, 223, 23, 122, 0`

---

{button ,AL("VENTURA\_Bitmap2D;;;;;0,"Defaultoverview",)} [Related Topics](#)

## .Bitmap2DPixelate (VENTURA)

**.Bitmap2DPixelate .Width=*long*, .Height=*long*, .Opacity=*long*, .Mode=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Pixelate filter to give your bitmap a blocky appearance or a circular, spiderweb look.

Syntax	Description
<b>.Width</b>	Specifies the width of the cells. Valid values range from 1 to 100.
<b>.Height</b>	Specifies the height of the cells. Valid values range from 1 to 100.
<b>.Opacity</b>	Specifies the opacity of the cells. Valid values range from 1 to 100%.
<b>.Mode</b>	Specifies the pixelation mode: 0 Square 1 Rectangular 2 Circular

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap2DPixelate 6, 25, 87, 2
```

---

{button ,AL('VENTURA\_Bitmap2D;;;;','0,"Defaultoverview",')} [Related Topics](#)

## .Bitmap2DPuzzle (VENTURA)

**.Bitmap2DPuzzle .Width=long, .Height=long, .Offset=long, .Fill=long, .RandSeed=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long**

This command applies a bitmap effect to a selected bitmap. The Puzzle filter breaks your bitmap down into puzzle-like pieces. You can set the size of the puzzle pieces, the distance between the pieces, and the color of the background.

Syntax	Description
.Width	Specifies the width of the puzzle blocks. Valid values range from 1 to 100.
.Height	Specifies the height of the puzzle blocks. Valid values range from 1 to 100.
.Offset	Specifies the amount of shifting that occurs. Valid values range from 0 to 200%.
.Fill	Specifies the method used to fill the empty area behind the puzzle pieces: 0 Black fill 1 White fill 2 Fill using the current paint color 3 Use the original image 4 Use a negative of the original image
.RandSeed	Provides a seed for a random number. You can use any value in this parameter.
.ColorModel	Specifies the color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.Bitmap2DPuzzle 20, 14, 40, 2, 185185452, 5, 239, 156, 159, 0`

---

{button ,AL("VENTURA\_Bitmap2D;;;;",0,"Defaultoverview",)} [Related Topics](#)

## .Bitmap2DRipple (VENTURA)

**.Bitmap2DRipple .Period=*long*, .Amplitude=*long*, .Angle=*long*, .Distort=*Boolean*, .Mode=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Ripple filter to make the surface of your bitmap appear like rippled waves of water. Apply the effect of a single wave ripple or the effect of two waves coming from different directions.

Syntax	Description
.Period	Specifies the distance between each wave cycle. Valid values range from 1 to 100.
.Amplitude	Specifies the amount of displacement created by each wave. Valid values range from 1 to 100.
.Angle	Specifies the direction of the ripple effect. Valid values range from 0 to 180 degrees.
.Distort	Set to TRUE (-1) to apply distortion to the ripple.
.Mode	Specifies the Ripple mode to use for the effect: 0 Single Wave 1 Dual Wave 1:1 2 Dual Wave 2:1

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap2DRipple 36, 11, 128, FALSE, 1
```

---

{button ,AL("VENTURA\_Bitmap2D;;;;";0,"Defaultoverview",)} [Related Topics](#)

# .Bitmap2DShear (VENTURA)


**.Bitmap2DShear .Scale=*long*, .Border=*long*, .Orientation=*long***


This command applies a bitmap effect to a selected bitmap. You can use the Shear filter to distort a bitmap along a path that can be customized, or you can use one of the preset shear curves. You can also use customized Shear filters.

The User Defined effect requires a block of commands. The block must begin with the **Bitmap2DShear** command and end with the **Bitmap2DEndShear** command. In between these two commands can be multiple **Bitmap2DShearTable** commands.

Syntax	Description
<b>.Scale</b>	Specifies the degree to which the image conforms to the curve. Set the value at 100% to have the image conform completely to the curve. Valid values range from 0 to 100 (%).
<b>.Border</b>	Specifies the method to use when filling the empty space left by the shear command: 0 Wraps the opposite end of the image around to the empty space 1 Stretches the image to fill the space 2 Fills empty areas with the current paint color
<b>.Orientation</b>	Specifies the direction of the shear curve: 0 Horizontal 1 Vertical

**Note**

 This command was introduced in Corel VENTURA 8.

 Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

```
.Bitmap2DShear 50, 2, 1, 5, 220, 43, 25, 0
.Bitmap2DShearTable 0, 512
.Bitmap2DShearTable 1, 515
.Bitmap2DShearTable 2, 517
.Bitmap2DShearTable 3, 520
.Bitmap2DShearTable 4, 523
... '(more .Bitmap2DShearTable command)
... '(more .Bitmap2DShearTable command)
... '(more .Bitmap2DShearTable command)
.Bitmap2DShearTable 1019, 501
.Bitmap2DShearTable 1020, 504
.Bitmap2DShearTable 1021, 507
.Bitmap2DShearTable 1022, 509
.Bitmap2DShearTable 1023, 512
.Bitmap2DEndShear
```

# .Bitmap2DShearTable (VENTURA)


**.Bitmap2DShearTable .Number=*long*, .Value=*long***


This command applies a bitmap effect to a selected bitmap. You can use the Shear filter to distort a bitmap along a path that can be customized, or you can use one of the preset shear curves. You can also use customized Shear filters.

The User Defined effect requires a block of commands. The block must begin with the **Bitmap2DShear** command and end with the **Bitmap2DEndShear** command. In between these two commands can be multiple **Bitmap2DShearTable** commands.

Syntax	Description
<b>.Number</b>	Specifies the index number of the point on the shear curve. Valid values range from 0 to 1023.
<b>.Value</b>	Specifies the displacement of the shear point. Valid values range from 0 to 1023.

**Note**

 This command was introduced in Corel VENTURA 8.

 Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

```
.Bitmap2DShear 50, 2, 1, 5, 220, 43, 25, 0
  .Bitmap2DShearTable 0, 512
  .Bitmap2DShearTable 1, 515
  .Bitmap2DShearTable 2, 517
  .Bitmap2DShearTable 3, 520
  .Bitmap2DShearTable 4, 523
... '(more .Bitmap2DShearTable command)
... '(more .Bitmap2DShearTable command)
... '(more .Bitmap2DShearTable command)
  .Bitmap2DShearTable 1019, 501
  .Bitmap2DShearTable 1020, 504
  .Bitmap2DShearTable 1021, 507
  .Bitmap2DShearTable 1022, 509
  .Bitmap2DShearTable 1023, 512
.Bitmap2DEndShear
```



## .Bitmap2DSwirl (VENTURA)

**.Bitmap2DSwirl .Angle=*long*, .CenterX=*long*, .CenterY=*long***

This command applies a bitmap effect to a selected bitmap. The Swirl filter lets you set the intensity, direction, and position of the swirling effect you apply to your bitmap.

Syntax	Description
<b>.Angle</b>	Specifies the angle through which the swirl occurs. Values range from -3600 to 3600.
<b>.CenterX</b>	Specifies the horizontal position of the swirl's center.
<b>.CenterY</b>	Specifies the vertical position of the swirl's center.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap2DSwirl -856, 210, 204
```

---

{button ,AL('VENTURA\_Bitmap2D;;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.Bitmap2DTile (VENTURA)**

**.Bitmap2DTile .Horizontal=*long*, .Vertical=*long***

This command applies a bitmap effect to a selected bitmap. The Tile filter reproduces your bitmap as a series of tiles. This filter is especially useful for previewing how your bitmap will look as a tiled background for Web pages.

Syntax	Description
<b>.Horizontal</b>	Specifies the number of times the image appears along the horizontal axis.
<b>.Vertical</b>	Specifies the number of times the image appears along the vertical axis.

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

`.Bitmap2DTile 10, 24`

---

{button ,AL("VENTURA\_Bitmap2D;;;;";0,"Defaultoverview",)} [Related Topics](#)

# .Bitmap2DTraceContour (VENTURA)

**.Bitmap2DTraceContour .Level=*long*, .EdgeType=*long***

This command applies a bitmap effect to a selected bitmap. The Trace Contour filter creates edges of different intensities by tracing bitmap elements using the 16 colors of the standard VGA palette. For the clearest results, use the Trace Contour filter on high-contrast bitmaps, for example, bitmaps that contain text.

Syntax	Description
.Level	Specifies the brightness threshold for outlining. Valid values range from 1 to 255.
.EdgeType	Specifies the type of edges to trace: <div><div>0</div><div>Traces the areas of your image where the brightness levels of the pixels fall below the <b>.Level</b> value</div><div>1</div><div>Traces the areas of your image where the brightness level of the pixels exceeds the <b>.Level</b> value</div></div>

**Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.Bitmap2DTraceContour 80, 1`

# .Bitmap2DUserDefined (VENTURA)



**.Bitmap2DUserDefined .Divisor=*long*, .Offset=*long***

This command applies a bitmap effect to a selected bitmap. The User Defined filter lets you create your own blur, sharpen, and edge detect special effects, based on values you enter into a 5 X 5 matrix.

The User Defined effect requires a block of commands. The block must begin with the **Bitmap2DUserDefined** command and end with the **Bitmap2DEndUserDefined** command. In between these two commands can be multiple **Bitmap2DUserDefinedPoint** commands.

Syntax	Description
<b>.Divisor</b>	Specifies the divisor value. After the command multiplies each matrix value by the brightness value of the corresponding pixel, it adds the products together, and then divides the sum by the value you type in the Divisor box.
<b>.Offset</b>	Specifies the offset value. This is the value that will be added to the final pixel values just before the effect is applied.

### Note

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap2DUserDefined 6, 0
.Bitmap2DUserDefinedPoint 0, 0
.Bitmap2DUserDefinedPoint 1, 0
.Bitmap2DUserDefinedPoint 2, 0
.Bitmap2DUserDefinedPoint 3, 0
.Bitmap2DUserDefinedPoint 4, 0
.Bitmap2DUserDefinedPoint 5, 0
.Bitmap2DUserDefinedPoint 6, 3
.Bitmap2DUserDefinedPoint 7, -2
.Bitmap2DUserDefinedPoint 8, 3
.Bitmap2DUserDefinedPoint 9, 0
.Bitmap2DUserDefinedPoint 10, 0
.Bitmap2DUserDefinedPoint 11, -2
.Bitmap2DUserDefinedPoint 12, 2
.Bitmap2DUserDefinedPoint 13, -2
.Bitmap2DUserDefinedPoint 14, 0
.Bitmap2DUserDefinedPoint 15, 0
.Bitmap2DUserDefinedPoint 16, 3
.Bitmap2DUserDefinedPoint 17, -2
.Bitmap2DUserDefinedPoint 18, 3
.Bitmap2DUserDefinedPoint 19, 0
.Bitmap2DUserDefinedPoint 20, 0
.Bitmap2DUserDefinedPoint 21, 0
.Bitmap2DUserDefinedPoint 22, 0
.Bitmap2DUserDefinedPoint 23, 0
.Bitmap2DUserDefinedPoint 24, 0
.Bitmap2DEndUserDefined
```

# .Bitmap2DUserDefinedPoint (VENTURA)



**.Bitmap2DUserDefinedPoint .Index=*long*, .Value=*long***

This command applies a bitmap effect to a selected bitmap. The User Defined filter lets you create your own blur, sharpen, and edge detect special effects, based on values you enter into a 5 X 5 matrix.

The User Defined effect requires a block of commands. The block must begin with the **Bitmap2DUserDefined** command and end with the **Bitmap2DEndUserDefined** command. In between these two commands can be multiple **Bitmap2DUserDefinedPoint** commands.

Syntax	Description
.Index	Specifies the index of the matrix point to define. Valid values range from 0 to 24.
.Value	Specifies the value of the specified matrix entry. Valid values range from -999 to 999.

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

```
.Bitmap2DUserDefined 6, 0
  .Bitmap2DUserDefinedPoint 0, 0
  .Bitmap2DUserDefinedPoint 1, 0
  .Bitmap2DUserDefinedPoint 2, 0
  .Bitmap2DUserDefinedPoint 3, 0
  .Bitmap2DUserDefinedPoint 4, 0
  .Bitmap2DUserDefinedPoint 5, 0
  .Bitmap2DUserDefinedPoint 6, 3
  .Bitmap2DUserDefinedPoint 7, -2
  .Bitmap2DUserDefinedPoint 8, 3
  .Bitmap2DUserDefinedPoint 9, 0
  .Bitmap2DUserDefinedPoint 10, 0
  .Bitmap2DUserDefinedPoint 11, -2
  .Bitmap2DUserDefinedPoint 12, 2
  .Bitmap2DUserDefinedPoint 13, -2
  .Bitmap2DUserDefinedPoint 14, 0
  .Bitmap2DUserDefinedPoint 15, 0
  .Bitmap2DUserDefinedPoint 16, 3
  .Bitmap2DUserDefinedPoint 17, -2
  .Bitmap2DUserDefinedPoint 18, 3
  .Bitmap2DUserDefinedPoint 19, 0
  .Bitmap2DUserDefinedPoint 20, 0
  .Bitmap2DUserDefinedPoint 21, 0
  .Bitmap2DUserDefinedPoint 22, 0
  .Bitmap2DUserDefinedPoint 23, 0
  .Bitmap2DUserDefinedPoint 24, 0
.Bitmap2DEndUserDefined
```

## .Bitmap2DWetPaint (VENTURA)

**.Bitmap2DWetPaint .Wetness=*long*, .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. You can apply successive combinations of positive and negative wetness values to the same bitmap to produce some incredible effects. For example, if you apply a negative Wetness value to an object, it will appear to have a drop shadow that smears down the page.

Syntax	Description
<b>.Wetness</b>	Specifies the range of colors that drip. Negative values cause the dark colors to drip, positive values cause the light colors to drip. Valid values range from -50 to 50.
<b>.Percentage</b>	Specifies the size of the paint drip. Valid values range from 0 to 100 (%).

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap2DWetPaint 24, 87
```

---

{button ,AL("VENTURA\_Bitmap2D;;;;";0,"Defaultoverview",)} [Related Topics](#)

## .Bitmap2DWhirlpool (VENTURA)

**.Bitmap2DWhirlpool .Spacing=*long*, .Smear=*long*, .Twist=*long*, .Streak=*long*, .Warp=*Boolean***

This command applies a bitmap effect to a selected bitmap. The Wind filter makes your bitmap appear blurred, as if the surface were smeared by a strong breeze. You can control the strength and direction of the wind.

Syntax	Description
<b>.Spacing</b>	Specifies the spacing between swirls. Valid values range from 5 to 200.
<b>.Smear</b>	Specifies the length of the swirls. Valid values range from 3 to 30.
<b>.Twist</b>	Specifies the degree of curvature. Valid values range from 0 to 90.
<b>.Streak</b>	Specifies the intensity of the swirls. Valid values range from 0 to 100.
<b>.Warp</b>	Set to TRUE (-1) if you want the whirlpool effect to distort the image. Set to FALSE (0) if you want the effect to overlay the image.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.Bitmap2DWind 91, 51, 259`

---

{button ,AL("VENTURA\_Bitmap2D;;;;";0,"Defaultoverview",)} [Related Topics](#)

## .Bitmap2DWind (VENTURA)

**.Bitmap2DWind .Strength=*long*, .Opacity=*long*, .Direction=*long***

This command applies a bitmap effect to a selected bitmap. The Whirlpool filter applies a pattern of fluid lines over your bitmap. This filter is memory-intensive and can take a while to apply and preview. Try experimenting with lower values first, and gradually increase the values.

Syntax	Description
<b>.Strength</b>	Specifies the intensity of the effect. Valid values range from 0 to 100%.
<b>.Opacity</b>	Specifies the opacity of the effect. Valid values range from 1 to 100%.
<b>.Direction</b>	Specifies the direction of the blur. Valid values range from 0 to 360 degrees.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap2DWind 91, 51, 259
```

---

{button ,AL("VENTURA\_Bitmap2D;;;;";0,"Defaultoverview",)} [Related Topics](#)



**SPECIAL EFFECTS 3D EFFECT**

## .Bitmap3DEmboss (VENTURA)

**.Bitmap3DEmboss .Depth=long, .Level=long, .Direction=long, .Color=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long**

This command applies a bitmap effect to a selected bitmap. You can use the Emboss filter to transform your bitmap into a relief, making the details appear as ridges and crevices on a flat surface.

Syntax	Description
<b>.Depth</b>	Specifies the depth of the ridges and crevices in the relief.
<b>.Level</b>	Specifies the amount of background color the relief will contain.
<b>.Direction</b>	Specifies the angle at which the light hits the relief. Valid values range from 0 to 360.
<b>.Color</b>	Specifies the emboss color: 0 The original color 1 Gray 2 Black 3 The current paper color
<b>.ColorModel</b>	Specifies the color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.Bitmap3DEmboss 3, 225, 15, 1, 5, 255, 255, 255, 0`

---

{button ,AL("VENTURA\_Bitmap3D;;;;",0,"Defaultoverview",)} [Related Topics](#)

## .Bitmap3DEndMeshWarp (VENTURA)

### .Bitmap3DEndMeshWarp

This command applies a bitmap effect to a selected bitmap. The Mesh Warp filter distorts a bitmap according to the manipulation of nodes on a grid. You can save and delete Mesh Warp filter styles.

The 3D Mesh Wrap bitmap effect requires a block of commands. The block must begin with the **Bitmap3DMeshWarp** command and end with the **.Bitmap3DEndMeshWarp** command. In between these two commands **Bitmap3DMeshPoint** commands which specify each point on the grid.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.Bitmap3DMeshWarp 5, 5
.Bitmap3DMeshPoint 0, 0, 0, 0
.Bitmap3DMeshPoint 0, 1, 0.25, 0
.Bitmap3DMeshPoint 0, 2, 0.5, 0
.Bitmap3DMeshPoint 0, 3, 0.75, 0
.Bitmap3DMeshPoint 0, 4, 1, 0
.Bitmap3DMeshPoint 1, 0, 0, 0.25
.Bitmap3DMeshPoint 1, 1, 0.25, 0.25
.Bitmap3DMeshPoint 1, 2, 0.5, 0.25
.Bitmap3DMeshPoint 1, 3, 0.75, 0.25
.Bitmap3DMeshPoint 1, 4, 1, 0.25
.Bitmap3DMeshPoint 2, 0, 0, 0.5
.Bitmap3DMeshPoint 2, 1, 0.25, 0.5
.Bitmap3DMeshPoint 2, 2, 0.5, 0.5
.Bitmap3DMeshPoint 2, 3, 0.75, 0.5
.Bitmap3DMeshPoint 2, 4, 1, 0.5
.Bitmap3DMeshPoint 3, 0, 0, 0.75
.Bitmap3DMeshPoint 3, 1, 0.25, 0.75
.Bitmap3DMeshPoint 3, 2, 0.5, 0.75
.Bitmap3DMeshPoint 3, 3, 0.806667, 0.784247
.Bitmap3DMeshPoint 3, 4, 1, 0.75
.Bitmap3DMeshPoint 4, 0, 0, 1
.Bitmap3DMeshPoint 4, 1, 0.25, 1
.Bitmap3DMeshPoint 4, 2, 0.5, 1
.Bitmap3DMeshPoint 4, 3, 0.75, 1
.Bitmap3DMeshPoint 4, 4, 1, 1
.Bitmap3DEndMeshWarp
```

---

{button ,AL('VENTURA\_Bitmap3D;;;','0,"Defaultoverview",)} [Related Topics](#)



# .Bitmap3DMapToObject (VENTURA)

**.Bitmap3DMapToObject .Mode=*long*, .Percentage=*long*, .Quality=*long***

This command applies a bitmap effect to a selected bitmap. The Map To Object filter lets you create exciting visual effects by wrapping your bitmap around a three-dimensional object.

Syntax	Description
.Mode	Specifies the mapping mode: 0 Spherical 1 Horizontal cylinder 2 Vertical cylinder
.Percentage	Specifies the direction and amount of wrapping. Negative percentage values wrap the image toward the back; positive percentage values wrap the image toward the front. Valid values range from -100 to 100 (%).
.Quality	Provides a list of preset quality levels that you can use when applying the filter: 0 Draft 1 Better 2 Best

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.Bitmap3DMapToObject 0, 36, 0`

---

`{button ,AL("VENTURA_Bitmap3D;;;;";0,"Defaultoverview",)}` [Related Topics](#)

# .Bitmap3DMeshPoint (VENTURA)



**.Bitmap3DMeshPoint .Row=long, .Column=long, .X=single, .Y=single**

This command applies a bitmap effect to a selected bitmap. The Mesh Warp filter distorts a bitmap according to the manipulation of nodes on a grid. You can save and delete Mesh Warp filter styles.

The 3D Mesh Wrap bitmap effect requires a block of commands. The block must begin with the **Bitmap3DMeshWarp** command and end with the **.Bitmap3DEndMeshWarp** command. In between these two commands **Bitmap3DMeshPoint** commands which specify each point on the grid.

Syntax	Description
<b>.Row</b>	Specifies the row number of the point you are defining.
<b>.Column</b>	Specifies the column number of the point you are defining.
<b>.X</b>	Specifies the horizontal coordinate of the point. Valid values range from 0.00 to 1.00.
<b>.Y</b>	Specifies the vertical coordinate of the point. Valid values range from 0.00 to 1.00.

## Note

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

## Example

```
.Bitmap3DMeshWarp 5, 5
.Bitmap3DMeshPoint 0, 0, 0, 0
.Bitmap3DMeshPoint 0, 1, 0.25, 0
.Bitmap3DMeshPoint 0, 2, 0.5, 0
.Bitmap3DMeshPoint 0, 3, 0.75, 0
.Bitmap3DMeshPoint 0, 4, 1, 0
.Bitmap3DMeshPoint 1, 0, 0, 0.25
.Bitmap3DMeshPoint 1, 1, 0.25, 0.25
.Bitmap3DMeshPoint 1, 2, 0.5, 0.25
.Bitmap3DMeshPoint 1, 3, 0.75, 0.25
.Bitmap3DMeshPoint 1, 4, 1, 0.25
.Bitmap3DMeshPoint 2, 0, 0, 0.5
.Bitmap3DMeshPoint 2, 1, 0.25, 0.5
.Bitmap3DMeshPoint 2, 2, 0.5, 0.5
.Bitmap3DMeshPoint 2, 3, 0.75, 0.5
.Bitmap3DMeshPoint 2, 4, 1, 0.5
.Bitmap3DMeshPoint 3, 0, 0, 0.75
.Bitmap3DMeshPoint 3, 1, 0.25, 0.75
.Bitmap3DMeshPoint 3, 2, 0.5, 0.75
.Bitmap3DMeshPoint 3, 3, 0.806667, 0.784247
.Bitmap3DMeshPoint 3, 4, 1, 0.75
.Bitmap3DMeshPoint 4, 0, 0, 1
.Bitmap3DMeshPoint 4, 1, 0.25, 1
.Bitmap3DMeshPoint 4, 2, 0.5, 1
.Bitmap3DMeshPoint 4, 3, 0.75, 1
.Bitmap3DMeshPoint 4, 4, 1, 1
.Bitmap3DEndMeshWarp
```

# .Bitmap3DMeshWarp (VENTURA)



**.Bitmap3DMeshWarp .Width=*long*, .Height=*long***

This command applies a bitmap effect to a selected bitmap. The Mesh Warp filter distorts a bitmap according to the manipulation of nodes on a grid. You can save and delete Mesh Warp filter styles.

The 3D Mesh Wrap bitmap effect requires a block of commands. The block must begin with the **Bitmap3DMeshWarp** command and end with the **.Bitmap3DEndMeshWarp** command. In between these two commands **Bitmap3DMeshPoint** commands which specify each point on the grid.

Syntax	Description
<b>.Width</b>	Specifies the number of columns in the warp grid. Valid values range from 5 to 11.
<b>.Height</b>	Specifies the number of rows in the warp grid. Valid values range from 5 to 11.

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

```
.Bitmap3DMeshWarp 5, 5
  .Bitmap3DMeshPoint 0, 0, 0, 0
  .Bitmap3DMeshPoint 0, 1, 0.25, 0
  .Bitmap3DMeshPoint 0, 2, 0.5, 0
  .Bitmap3DMeshPoint 0, 3, 0.75, 0
  .Bitmap3DMeshPoint 0, 4, 1, 0
  .Bitmap3DMeshPoint 1, 0, 0, 0.25
  .Bitmap3DMeshPoint 1, 1, 0.25, 0.25
  .Bitmap3DMeshPoint 1, 2, 0.5, 0.25
  .Bitmap3DMeshPoint 1, 3, 0.75, 0.25
  .Bitmap3DMeshPoint 1, 4, 1, 0.25
  .Bitmap3DMeshPoint 2, 0, 0, 0.5
  .Bitmap3DMeshPoint 2, 1, 0.25, 0.5
  .Bitmap3DMeshPoint 2, 2, 0.5, 0.5
  .Bitmap3DMeshPoint 2, 3, 0.75, 0.5
  .Bitmap3DMeshPoint 2, 4, 1, 0.5
  .Bitmap3DMeshPoint 3, 0, 0, 0.75
  .Bitmap3DMeshPoint 3, 1, 0.25, 0.75
  .Bitmap3DMeshPoint 3, 2, 0.5, 0.75
  .Bitmap3DMeshPoint 3, 3, 0.806667, 0.784247
  .Bitmap3DMeshPoint 3, 4, 1, 0.75
  .Bitmap3DMeshPoint 4, 0, 0, 1
  .Bitmap3DMeshPoint 4, 1, 0.25, 1
  .Bitmap3DMeshPoint 4, 2, 0.5, 1
  .Bitmap3DMeshPoint 4, 3, 0.75, 1
  .Bitmap3DMeshPoint 4, 4, 1, 1
.Bitmap3DEndMeshWarp
```

## .Bitmap3DPageCurl (VENTURA)

**.Bitmap3DPageCurl .Corner=*long*, .Direction=*Boolean*, .Width=*long*, .Height=*long*, .Opaque=*Boolean*, .CurlColorModel=*long*, .CurlColor1=*long*, .CurlColor2=*long*, .CurlColor3=*long*, .CurlColor4=*long*, .BackColorModel=*long*, .BackColor1=*long*, .BackColor2=*long*, .BackColor3=*long*, .BackColor4=*long***

This command applies a bitmap effect to a selected bitmap. The Page Curl filter is used to give the impression that a corner of your bitmap has rolled in on itself.

Syntax	Description
<b>.Corner</b>	Specifies the corner to curl: 0 = Top-left 1 = Top-right 2 = Bottom-left 3 = Bottom-right
<b>.Direction</b>	Set to TRUE (-1) to have the page curl begin along the top or bottom edge of the image, depending on the .Corner location. Set to FALSE (0) to have the page curl begin along the left or right edge of the image.
<b>.Width</b>	Specifies the width of the page curl. Increase the value to extend the page curl along the vertical edge of the image.
<b>.Height</b>	Specifies the height of the page curl. Increase the value to extend the page curl along the horizontal edge of the image.
<b>.Opaque</b>	Set to TRUE (-1) to make the curl completely opaque. Set to FALSE (0) to have some of the image show through the curl.
<b>.CurlColorModel</b>	Specifies the color model to use for the curl color: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.CurlColor1</b>	Specifies the first color component for the <b>.CurlColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.CurlColor2</b>	Specifies the second color component for the <b>.CurlColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.CurlColor3</b>	Specifies the third color component for the <b>.CurlColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.CurlColor4</b>	Specifies the fourth color component for the <b>.CurlColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.BackColorModel</b>	Specifies the color model to use for the back color: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a>

- 12 [L\\*a\\*b\\*](#)
- 14 [PANTONE Hexachrome](#)

- .BackColor1** Specifies the first color component for the **.BackColorModel** parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
- .BackColor2** Specifies the second color component for the **.BackColorModel** parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
- .BackColor3** Specifies the third color component for the **.BackColorModel** parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
- .BackColor4** Specifies the fourth color component for the **.BackColorModel** parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

**Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

```
.Bitmap3DPageCurl 2, FALSE, 57, 40, FALSE, 5, 127, 127, 127, 0, 5, 220, 43, 25, 0
```

---

{button ,AL('VENTURA\_Bitmap3D;;;;','0,"Defaultoverview",)} [Related Topics](#)





# .Bitmap3DPerspective (VENTURA)

**.Bitmap3DPerspective .X1=single, .Y1=single, .X2=single, .Y2=single, .X3=single, .Y3=single, .X4=single, .Y4=single, .Shear=Boolean**

This command applies a bitmap effect to a selected bitmap. The Perspective filter changes the perspective of your bitmap, giving it a three-dimensional look.

Syntax	Description
.X1	The horizontal coordinate of the top-left handle of the distortion rectangle. Valid values range from 0 to 48.
.Y1	The vertical coordinate of the top-left handle of the distortion rectangle. Valid values range from 0 to 48.
.X2	The horizontal coordinate of the top-right handle of the distortion rectangle. Valid values range from 0 to 48.
.Y2	The vertical coordinate of the top-right handle of the distortion rectangle. Valid values range from 0 to 48.
.X3	The horizontal coordinate of the bottom-right handle of the distortion rectangle. Valid values range from 0 to 48.
.Y3	The vertical coordinate of the bottom-right handle of the distortion rectangle. Valid values range from 0 to 48.
.X4	The horizontal coordinate of the bottom-left handle of the distortion rectangle. Valid values range from 0 to 48.
.Y4	The vertical coordinate of the bottom-left handle of the distortion rectangle. Valid values range from 0 to 48.
.Shear	Set to TRUE (-1) to keep two nodes equidistant at all times.

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.Bitmap3DPerspective 0, 0, 48, 23.5, 48, 70.2458, 0, 46.7458, TRUE`

## .Bitmap3DPinchPunch (VENTURA)

**.Bitmap3DPinchPunch .Level=*long*, .CenterX=*long*, .CenterY=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Pinch/Punch filter to warp your bitmap in three-dimensions by either "pinching" the bitmap away from you, or "punching" it toward you.

Syntax	Description
<b>.Level</b>	Specifies the intensity of the pinch or punch effect. Positive values apply a pinch effect, while negative values apply a punch effect. Valid values range from -100 to 100.
<b>.CenterX</b>	Specifies the horizontal position of the pinch or punch's center.
<b>.CenterY</b>	Specifies the vertical position of the pinch or punch's center.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap3DPinchPunch -58, 210, 204
```

---

{button ,AL("VENTURA\_Bitmap3D;;;;",0,"Defaultoverview",)} [Related Topics](#)

## .Bitmap3DRotate (VENTURA)

**.Bitmap3DRotate .Horizontal=*long*, .Vertical=*long*, .Face=*long*, .BestFit=*Boolean***

This command applies a bitmap effect to a selected bitmap. You can use the 3D Rotate filter to rotate your bitmap as if it were one side of a three-dimensional box. The shaded side of the box represents your bitmap.

Syntax	Description
<b>.Horizontal</b>	Specifies the degree of horizontal rotation. Valid values range from -75 to 75.
<b>.Vertical</b>	Specifies the degree of vertical rotation. Valid values range from -75 to 75.
<b>.Face</b>	Specifies the face of the rotation cube that faces forward. Valid values range from 0 to 4.
<b>.BestFit</b>	Set to TRUE (-1) if you want to ensure that all parts of your image remain within the Image Window.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.Bitmap3DRotate 28, -2, 0, TRUE
```

---

{button ,AL('VENTURA\_Bitmap3D;;;;',0,"Defaultoverview",)} [Related Topics](#)



# .Bitmap3DZigZag (VENTURA)

**.Bitmap3DZigZag .Waves=*long*, .Strength=*long*, .Damping=*long*, .Type=*long*, .CenterX=*long*, .CenterY=*long***

This command applies a bitmap effect to a selected bitmap. You can create a variety of dramatic effects using the Zig Zag filter, for example, you can make your bitmap appear twisted in a series of concentric circles.

Syntax	Description
.Waves	Specifies the period of the waves. Valid values range from 1 to 100.
.Strength	Specifies the intensity of the distortion. Valid values range from 1 to 100.
.Damping	Specifies the degree of damping in successive waves. Large values cause the distortion waves to phase out toward the edges of your image, Smaller values cause the waves to extend toward the edges. Valid values range from 1 to 100.
.Type	Specifies the type of wave: 0      Pond ripples 1      Out from center 2      Around center
.CenterX	Specifies the horizontal position of the zig zag effect.
.CenterY	Specifies the vertical position of the zig zag effect.

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.Bitmap3DZigZag 14, 30, 80, 2, 210, 204`

## **SPECIAL EFFECTS ARTISTIC**

## .BitmapArtisticCanvas (VENTURA)

**.BitmapArtisticCanvas** *.FileName=string, .Transparency=long, .Emboss=long, .X=long, .Y=long, .Mode=long, .Offset=long*

This command applies a bitmap effect to a selected bitmap. The Canvas filter gives the surface of your bitmap an embossed, textured look. For best results, use a canvas map that has highly contrasting features.

Syntax	Description
<b>.Filename</b>	Specifies the name and path of the file to use as the canvas surface.
<b>.Transparency</b>	Specifies the transparency of the canvas effect. Valid values range from 0 to 100 (%).
<b>.Emboss</b>	Specifies the degree of embossing. Embossing gives the canvas a raised, relief effect. Valid values range from 0 to 100 (%).
<b>.X</b>	Specifies the horizontal offset of the canvas map. Valid values range from 0 to 100 (%).
<b>.Y</b>	Specifies the vertical offset of the canvas map. Valid values range from 0 to 100 (%).
<b>.Mode</b>	Specifies the offset mode: 0 Rows. Offsets rows of tiles 1 Columns. Offsets columns of tiles 2 Stretch To Fit
<b>.Offset</b>	Specifies the degree of offset. Valid values range from 0 to 100 (%).

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapArtisticCanvas "c:\canvas\linen2c.pcx", 91, 93, 9, -9, 0, 0
```

---

{button ,AL('VENTURA\_BitmapArtistic;;;;','0,"Defaultoverview",)} Related Topics

## .BitmapArtisticGlassBlock (VENTURA)

**.BitmapArtisticGlassBlock .Width=*long*, .Height=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Glass Block filter to give your bitmap the effect of being viewed through a number of glass blocks. For best results, use mid-range block sizes.

Syntax	Description
<b>.Width</b>	Specifies the width of the glass blocks. Valid values range from 1 to 100.
<b>.Height</b>	Specifies the height of the glass blocks. Valid values range from 1 to 100.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapArtisticGlassBlock 51, 66
```

---

{button ,AL("VENTURA\_BitmapArtistic;;;;";'0,"Defaultoverview",)} [Related Topics](#)

## .BitmapArtisticImpressionist (VENTURA)

**.BitmapArtisticImpressionist .Horizontal=*long*, .Vertical=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Impressionist filter to make your bitmap look like an Impressionist painting.

Syntax	Description
<b>.Horizontal</b>	Specifies the amount of pixel displacement that occurs along the horizontal axis. Valid values range from 1 to 100.
<b>.Vertical</b>	Specifies the amount of pixel displacement that occurs along the horizontal axis. Valid values range from 1 to 100.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapArtisticImpressionist 10, 33
```

---

{button ,AL("VENTURA\_BitmapArtistic;;;;",'0,"Defaultoverview"),} [Related Topics](#)



## .BitmapArtisticSmokedGlass (VENTURA)

**.BitmapArtisticSmokedGlass .Tint=long, .Percent=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long**

This command applies a bitmap effect to a selected bitmap. The Smoked Glass filter places a colored tint over your bitmap, like a sheet of colored glass. You can control the opacity and blurriness of the glass.

Syntax	Description
<b>.Tint</b>	Specifies the opacity of the tint. Valid values range from 0 to 100 (%).
<b>.Percent</b>	Specifies the amount of blurring to use to create the glass effect. Valid values range from 0 to 100 (%).
<b>.ColorModel</b>	Specifies the color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapArtisticSmokedGlass 40, 70, 5, 223, 23, 122, 0
```

---

{button ,AL('VENTURA\_BitmapArtistic;;;;','0','Defaultoverview',)} [Related Topics](#)

## .BitmapArtisticVignette (VENTURA)

**.BitmapArtisticVignette .Shape=*long*, .Offset=*long*, .Fade=*long*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long***

This command applies a bitmap effect to a selected bitmap. Add professional-looking framing effects to your bitmaps with the Vignette filter. You can set the shape, color, and fade rate of the frames.

Syntax	Description
<b>.Shape</b>	Specifies the frame shape: 0 Elliptical 1 Circular 2 Rectangular 3 Square
<b>.Offset</b>	Specifies the size of the frame. Valid values range from 0 to 140.
<b>.Fade</b>	Specifies the fade rate between the image and the frame. Valid values range from 0 to 100.
<b>.ColorModel</b>	Specifies the color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 RGB (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapArtisticVignette 1, 59, 80, 5, 223, 23, 122, 0
```

---

{button ,AL('VENTURA\_BitmapArtistic;;;;',0,"Defaultoverview",)} [Related Topics](#)

## **SPECIAL EFFECTS BLUR**

## **.BitmapBlurDirectionalSmooth (VENTURA)**

**.BitmapBlurDirectionalSmooth .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. The Directional Smooth filter applies a very subtle amount of blurring to your bitmap, so that the bitmap isn't distorted.

### **Syntax**

### **Description**

**.Percentage**

Specifies the intensity of the effect. Valid values range from 0 to 100 (%).

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

`.BitmapBlurDirectionalSmooth 89`

---

{button ,AL('VENTURA\_BitmapBlur;;;;','0',"Defaultoverview",)} [Related Topics](#)

## .BitmapBlurGaussian (VENTURA)

**.BitmapBlurGaussian .Radius=*long***

This command applies a bitmap effect to a selected bitmap. The Gaussian Blur filter produces a hazy effect, blurring the bitmap according to a Gaussian distribution, which spreads the pixel information outward using bell-shaped curves.

Syntax	Description
<b>.Radius</b>	Specifies the intensity of the effect. Valid values range from 1 to 50.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.BitmapBlurGaussian 3036`

---

{button ,AL('VENTURA\_BitmapBlur;;;;','0',"Defaultoverview",)} [Related Topics](#)

## **.BitmapBlurJaggy (VENTURA)**

**.BitmapBlurJaggy .Width=*long*, .Height=*long***

This command applies a bitmap effect to a selected bitmap. The Jaggy Despeckle filter applies a soft, blurred effect to your bitmap. It is especially effective on bitmaps with a high amount of contrast.

Syntax	Description
<b>.Width</b>	Specifies the intensity of horizontal color scattering. Valid values range from 1 to 5.
<b>.Height</b>	Specifies the intensity of vertical color scattering. Valid values range from 1 to 5.

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

`.BitmapBlurJaggy 2, 3`

---

`{button ,AL('VENTURA_BitmapBlur;;;;','0,"Defaultoverview",)} Related Topics`

## .BitmapBlurLowPass (VENTURA)

**.BitmapBlurLowPass .Radius=*long*, .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. Use the Low Pass filter to remove sharp edges and detail from your bitmap. High settings will erase much of the bitmap's detail.

Syntax	Description
<b>.Radius</b>	Specifies the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
<b>.Percentage</b>	Specifies the intensity of the effect. Larger values reduce harsh transitions between shadows and highlights. Valid values range from 0 to 100 (%).

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapBlurLowPass 6, 64
```

---

{button ,AL("VENTURA\_BitmapBlur;;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.BitmapBlurMotion (VENTURA)**

**.BitmapBlurMotion .Speed=*long*, .Direction=*long*, .Method=*long***

This command applies a bitmap effect to a selected bitmap. The Motion Blur filter blurs your bitmap like a photograph of a moving object.

Syntax	Description
<b>.Speed</b>	Specifies the degree of image blurring. Valid values range from 1 to 999 pixels.
<b>.Direction</b>	Specifies the direction of blurring. Valid values range from 0 to 360 degrees.
<b>.Method</b>	Specifies the method of off-image sampling: 0 Ignores pixels outside the image 1 Uses paper color 2 Samples nearest edge pixel

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

```
.BitmapBlurMotion 25, 120, 2
```

---

{button ,AL('VENTURA\_BitmapBlur;;;;','0',"Defaultoverview"),} [Related Topics](#)





# .BitmapBlurRadial (VENTURA)

**.BitmapBlurRadial .Mode=*long*, .Quality=*long*, .Amount=*long*, .CenterX=*long*, .CenterY=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Radial Blur filter to give your bitmap a blurred effect that radiates out from a central point. Choose the fast quality level to speed up the effect.

Syntax	Description
<b>.Mode</b>	Specifies the radial mode: 0 Spin 1 Zoom
<b>.Quality</b>	Specifies the quality: 0 Best 1 Fast
<b>.Amount</b>	Specifies the radius of the blur. Valid values range from 0 to 100.
<b>.CenterX</b>	Specifies the horizontal position of the blur's center.
<b>.CenterY</b>	Specifies the vertical position of the blur's center.

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.BitmapBlurRadial 1, 31, 1, 210, 204`

---

{button ,AL('VENTURA\_BitmapBlur;;;;','0,"Defaultoverview",)} [Related Topics](#)



# .BitmapBlurSmooth (VENTURA)

**.BitmapBlurSmooth .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. The Smooth filter applies an extremely subtle amount of blurring to your bitmap that may only be apparent at high zoom levels. Unlike the Directional Smooth filter, which blurs along the edges of your bitmap, the Smooth filter blurs all pixels equally.

Syntax	Description
<b>.Percentage</b>	Specifies the intensity of the effect. Valid values range from 0 to 100 (%).

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.BitmapBlurSmooth 82`

## **.BitmapBlurSoften (VENTURA)**

**.BitmapBlurSoften .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. The Soften filter slightly blurs your bitmap, retaining a high level of detail. Try using the Smooth filter for a similar effect.

Syntax	Description
<b>.Percentage</b>	Specifies the intensity of the effect. Valid values range from 0 to 100 (%).

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

`.BitmapBlurSoften 26`

---

{button ,AL('VENTURA\_BitmapBlur;;;;','0',"Defaultoverview",)} [Related Topics](#)

## **SPECIAL EFFECTS COLOR TRANSFORM**

## .BitmapColorTransformBitPlanes (VENTURA)

**.BitmapColorTransformBitPlanes .Red=*long*, .Green=*long*, .Blue=*long***

This command applies a bitmap effect to a selected bitmap. The Bit Planes filter reduces the bitmap to basic RGB color components and represents the tonal changes of your bitmap with solid areas of color. It is particularly useful for analyzing bitmap gradients.

Syntax	Description
<b>.Red</b>	Specifies the color sensitivity in the red channel. Valid values range from 0 to 7.
<b>.Green</b>	Specifies the color sensitivity in the green channel. Valid values range from 0 to 7.
<b>.Blue</b>	Specifies the color sensitivity in the blue channel. Valid values range from 0 to 7.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapColorTransformBitPlanes 4, 2, 3
```

---

{button ,AL("VENTURA\_BitmapColorTransform;;;;;','0,"Defaultoverview"),} [Related Topics](#)

## .BitmapColorTransformHalftone (VENTURA)

**.BitmapColorTransformHalftone .Radius=*long*, .Cyan=*long*, .Magenta=*long*, .Yellow=*long*, .Black=*long***

This command applies a bitmap effect to a selected bitmap. Use this filter to give your bitmap the appearance of a color halftone.

Syntax	Description
<b>.Radius</b>	Specifies the dot radius. Valid values range from 2 to 10.
<b>.Cyan</b>	Specifies the Cyan channel angle. Valid values range from 0 to 359 degrees.
<b>.Magenta</b>	Specifies the Magenta channel angle. Valid values range from 0 to 359 degrees.
<b>.Yellow</b>	Specifies the Yellow channel angle. Valid values range from 0 to 359 degrees.
<b>.Black</b>	Specifies the Black channel angle. Valid values range from 0 to 359 degrees.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapColorTransformHalftone 7, 47, 42, 35, 45
```

---

{button ,AL("VENTURA\_BitmapColorTransform;;;;;','0,"Defaultoverview"),} [Related Topics](#)

## .BitmapColorTransformPsychedelic (VENTURA)

**.BitmapColorTransformPsychedelic .Level=*long***

This command applies a bitmap effect to a selected bitmap. The Psychedelic filter transforms the colors of your bitmap into shocking, bright colors. Small changes to the Level slider setting make a great difference.

Syntax	Description
<b>.Level</b>	Specifies the intensity of the effect. Valid values range from 0 to 255.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapColorTransformPsychedelic 175
```

---

{button ,AL('VENTURA\_BitmapColorTransform;;;;','0','Defaultoverview'),} [Related Topics](#)

## **.BitmapColorTransformSolarize (VENTURA)**

**.BitmapColorTransformSolarize .Level=*long***

This command applies a bitmap effect to a selected bitmap. The Solarize filter makes your bitmap look like a negative photographic image.

Syntax	Description
<b>.Level</b>	Specifies the intensity of the effect. Valid values range from 0 to 255.

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

```
.BitmapColorTransformSolarize 79
```

---

{button ,AL('VENTURA\_BitmapColorTransform;;;;','0,"Defaultoverview",)} [Related Topics](#)



## **SPECIAL EFFECTS NOISE**



# .BitmapNoiseAdd (VENTURA)

**.BitmapNoiseAdd .Level=long, .Density=long, .ColorNoise=Boolean, .NoiseType=long**

This command applies a bitmap effect to a selected bitmap. Use the Add Noise filter to create a granular effect that adds random pixels across your bitmap. You can choose from three different types of noise, set the amount of noise, and apply pixels of different colors.

Syntax	Description
.Level	Specifies how much noise to add. Valid values range from 0 to 100.
.Density	Specifies the density of noise addition. Valid values range from 0 to 100.
.ColorNoise	Set to TRUE (-1) to apply randomly colored noise to the image. Set to FALSE (0) for monochrome noise.
.NoiseType	Specifies the type of noise: 0 Gaussian. Prioritizes colors along a Gaussian curve 1 Spike. Uses colors that are distributed around a narrow curve 2 Uniform. Provides an overall granular appearance

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.BitmapNoiseAdd 80, 41, TRUE, 1`

---

{button ,AL('VENTURA\_BitmapNoise;;;;','0,"Defaultoverview",)} Related Topics

## .BitmapNoiseDiffuse (VENTURA)

**.BitmapNoiseDiffuse .Level=*long***

This command applies a bitmap effect to a selected bitmap. The Diffuse filter removes noise by spreading out the pixels of your bitmap to fill in blank spaces. The intensity of the effect can vary quite noticeably, depending on the Level slider setting you choose.

Syntax	Description
.Level	Specifies the intensity of the effect. Valid values range from 0 to 100 (%).

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapNoiseDiffuse 68
```

---

{button ,AL("VENTURA\_BitmapNoise;;;;','0,"Defaultoverview",)} [Related Topics](#)

## .BitmapNoiseDustScratch (VENTURA)

**.BitmapNoiseDustScratch .Threshold=*long*, .Radius=*long***

This command applies a bitmap effect to a selected bitmap. The Dust and Scratch filter lets you repair damage to an image. It reduces bitmap noise by averaging pixel values.

Syntax	Description
<b>.Threshold</b>	Specifies how great a change in value must occur to any pixel before the effect is applied. Valid values range from 0 to 255.
<b>.Radius</b>	Specifies the range of the effect. Larger values increase the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapNoiseDustScratch 40, 8
```

---

{button ,AL('VENTURA\_BitmapNoise;;;;','0,"Defaultoverview",)} [Related Topics](#)

## .BitmapNoiseMaximum (VENTURA)

**.BitmapNoiseMaximum .Radius=*long*, .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. Use the Maximum filter to remove noise from your bitmap.

Syntax	Description
<b>.Radius</b>	Specifies the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
<b>.Percentage</b>	Specifies the intensity of the effect. Valid values range from 0 to 100 (%).

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapNoiseMaximum 6, 79
```

---

{button ,AL("VENTURA\_BitmapNoise;;;;','0,"Defaultoverview",)} [Related Topics](#)

## .BitmapNoiseMedian (VENTURA)

**.BitmapNoiseMedian .Radius=*long***

This command applies a bitmap effect to a selected bitmap. The Median filter removes noise from your bitmap by blurring neighboring pixels.

Syntax	Description
<b>.Radius</b>	Specifies the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.BitmapNoiseMedian 5`

---

{button ,AL('VENTURA\_BitmapNoise;;;;','0,"Defaultoverview",)} [Related Topics](#)

## .BitmapNoiseMinimum (VENTURA)

**.BitmapNoiseMinimum .Radius=*long*, .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. The Minimum filter removes noise by darkening the pixels of your bitmap.

Syntax	Description
.Radius	Specifies the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
.Percentage	Specifies the intensity of the effect. Valid values range from 0 to 100 (%).

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

`.BitmapNoiseMinimum 7, 90`

---

{button ,AL('VENTURA\_BitmapNoise;;;;','0,"Defaultoverview",)} [Related Topics](#)

## .BitmapNoiseRemove (VENTURA)

**.BitmapNoiseRemove** .Threshold=*long*, .Auto=*Boolean*

This command applies a bitmap effect to a selected bitmap. Use the Remove Noise filter to soften your bitmap and to remove random pixel noise.

Syntax	Description
.Threshold	Specifies how great a change in value must occur to any pixel before the effect is applied.
.Auto	Set to TRUE (-1) to have Corel VENTURA automatically calculate the noise reduction level required to improve image quality. Set to FALSE (0) to use the threshold value above.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapNoiseRemove 61, FALSE
```

---

{button ,AL('VENTURA\_BitmapNoise;;;;','0,"Defaultoverview",)} [Related Topics](#)



# .BitmapNoiseRemoveMoire (VENTURA)

**.BitmapNoiseRemoveMoire .Level=*long*, .Mode=*long*, .FinalRes=*long*, .OrgRes=*long***

This command applies a bitmap effect to a selected bitmap. Use the Remove Noise filter to soften your bitmap and to remove random pixel noise.

Syntax	Description
.Level	Specifies the intensity of the effect. Valid values range from 0 to 10.
.Mode	Specifies the level of the effect. 0 Better 1 Faster
.FinalRes	Specifies the resolution of the image after the filter is applied.
.OrgRes	Specifies the resolution of the image before the filter is applied.

## Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

## Example

`.BitmapNoiseRemoveMoire 8, 0`

**{button ,AL('VENTURA\_BitmapNoise;;;;','0,"Defaultoverview",)} Related Topics**

## **SPECIAL EFFECTS RENDER**

## .BitmapRender3DStereoNoise (VENTURA)

**.BitmapRender3DStereoNoise** *.Depth=long, .Dots=Boolean*

This command applies a bitmap effect to a selected bitmap. Use the 3D Stereo Noise filter to create a dithered noise pattern that has a three-dimensional appearance when viewed a certain way. For best results, use bitmaps with a high level of contrast.

Syntax	Description
<b>.Depth</b>	Specifies the depth of the stereogram image. Valid values range from 1 to 9.
<b>.Dots</b>	Set to TRUE (-1) to display dots to help focus on the stereogram image.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapRender3DStereoNoise 5, FALSE
```

---

{button ,AL("VENTURA\_BitmapRender;;;;','0,"Defaultoverview",)} [Related Topics](#)

## .BitmapRenderEndLighting (VENTURA)

### .BitmapRenderEndLighting

This command applies a bitmap effect to a selected bitmap. Add light sources, ambient lighting, or both to your bitmap with the Lighting Effects filter. You can control the color, brightness, and contrast of the light sources.

The Lighting Effects bitmap effect requires a block of commands. The block must begin with the **BitmapRenderLighting** command and end with the **BitmapRenderEndLighting** command. In between these two commands can be multiple **BitmapRenderLightingSource** commands.

#### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

#### Example

```
.BitmapRenderLighting 2
.BitmapRenderLightingSource 0, 0, 0, 0, 75, 0, 90, 201, 180, 100, 0, 50, 0, 25, 0, 255, 255,
255, TRUE
.BitmapRenderLightingSource 1, 0.0794293, 0.113127, 1, 75, 135, 45, 100, 90, 67, 30, 50, 0,
25, 0, 255, 255, 255, TRUE
.BitmapRenderEndLighting
```

The above example 2 to light sources to a selected bitmap.

---

{button ,AL("VENTURA\_BitmapRender;;;;','0,"Defaultoverview",)} [Related Topics](#)

## .BitmapRenderLensFlare (VENTURA)

**.BitmapRenderLensFlare** *.X=*single, *.Y=*single, *.Brightness=*long, *.LensType=*long, *.ColorModel=*long, *.Color1=*long, *.Color2=*long, *.Color3=*long, *.Color4=*long

This command applies a bitmap effect to a selected bitmap. The Lens Flare filter simulates bright light striking a camera lens, the flare created, as well as the bright flare point, is refracted into a series of small lightened circles.

Syntax	Description
<b>.X</b>	Specifies the horizontal coordinate of the flare's center.
<b>.Y</b>	Specifies the vertical coordinate of the flare's center.
<b>.Brightness</b>	Specifies the intensity of the lens flare. The effect of the brightness setting varies with different lens types.
<b>.LensType</b>	Specifies the type of lens, which will affect the appearance of the flare: 0 50-300mm zoom 1 35 mm prime 2 105 mm prime
<b>.ColorModel</b>	Specifies the color model to use: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapRenderLensFlare 0.698246, 0.435252, 158, 1, 5, 254, 248, 0, 0
```

---

{button ,AL('VENTURA\_BitmapRender;;;;',0,"Defaultoverview"),} [Related Topics](#)

# .BitmapRenderLighting (VENTURA)



.BitmapRenderLighting .Sources=*long*

This command applies a bitmap effect to a selected bitmap. Add light sources, ambient lighting, or both to your bitmap with the Lighting Effects filter. You can control the color, brightness, and contrast of the light sources.

The Lighting Effects bitmap effect requires a block of commands. The block must begin with the **BitmapRenderLighting** command and end with the **BitmapRenderEndLighting** command. In between these two commands can be multiple **BitmapRenderLightingSource** commands.

Syntax	Description
.Sources	Specifies the number of light sources to use.

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

```
.BitmapRenderLighting 2
.BitmapRenderLightingSource 0, 0, 0, 0, 75, 0, 90, 201, 180, 100, 0, 50, 0, 25, 0, 255, 255,
255, TRUE
.BitmapRenderLightingSource 1, 0.0794293, 0.113127, 1, 75, 135, 45, 100, 90, 67, 30, 50, 0,
25, 0, 255, 255, 255, TRUE
.BitmapRenderEndLighting
```

The above example 2 to light sources to a selected bitmap.

## .BitmapRenderLightingSource (VENTURA)

**.BitmapRenderLightingSource** *.Index=long, .X=single, .Y=single, .Type=long, .Height=long, .Direction=long, .Elevation=long, .Intensity=long, .Aperture=long, .Focus=long, .Whiteness=long, .Exposure=long, .Channel=long, .Depth=long, .Contrast=long, .Red=long, .Green=long, .Blue=long, .Switch=Boolean*

This command applies a bitmap effect to a selected bitmap. Add light sources, ambient lighting, or both to your bitmap with the Lighting Effects filter. You can control the color, brightness, and contrast of the light sources.

The Lighting Effects bitmap effect requires a block of commands. The block must begin with the **BitmapRenderLighting** command and end with the **BitmapRenderEndLighting** command. In between these two commands can be multiple **BitmapRenderLightingSource** commands.

Syntax	Description
<b>.Index</b>	Specifies the light source number. The first light source is 0, the second light source is 1, and so on.
<b>.X</b>	Specifies the horizontal coordinate for the light source.
<b>.Y</b>	Specifies the vertical coordinate for the light source.
<b>.Type</b>	Specifies the light source type.
<b>.Height</b>	Specifies the height of the light source.
<b>.Direction</b>	Specifies the direction of the light source.
<b>.Elevation</b>	Specifies the elevation of the light source.
<b>.Intensity</b>	Specifies the intensity of the light source.
<b>.Aperture</b>	Specifies the aperture of the light source.
<b>.Focus</b>	Specifies the focus of the light source.
<b>.Whiteness</b>	Specifies the whiteness of the light source.
<b>.Exposure</b>	Specifies the exposure.
<b>.Channel</b>	Specifies the channel.
<b>.Depth</b>	Specifies the depth.
<b>.Contrast</b>	Specifies the contrast.
<b>.Red</b>	Specifies the Red value.
<b>.Green</b>	Specifies the Green value.
<b>.Blue</b>	Specifies the Blue value.
<b>.Switch</b>	Specifies whether to switch. Set to TRUE (-1) to enable.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapRenderLighting 2
.BitmapRenderLightingSource 0, 0, 0, 0, 75, 0, 90, 201, 180, 100, 0, 50, 0, 25, 0, 255, 255,
255, TRUE
.BitmapRenderLightingSource 1, 0.0794293, 0.113127, 1, 75, 135, 45, 100, 90, 67, 30, 50, 0,
25, 0, 255, 255, 255, TRUE
.BitmapRenderEndLighting
```

The above example 2 to light sources to a selected bitmap.

---

{button ,AL('VENTURA\_BitmapRender;;;;','0','Defaultoverview',)} Related Topics

## **SPECIAL EFFECTS SHARPEN**



# .BitmapSharpen (VENTURA)

**.BitmapSharpen .EdgeLevel=*long*, .Background=*long***

This command applies a bitmap effect to a selected bitmap. The Sharpen filter accentuates edge detail by bringing blurred areas in your bitmap into focus. In addition, contrast between neighboring pixels is increased.

Syntax	Description
<b>.EdgeLevel</b>	Specifies the amount of edge sharpening. Valid values range from 0 to 100 (%).
<b>.Background</b>	Specifies how great a change in value must occur to any pixel before the effect is applied.

**Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.BitmapSharpen 71, 55`

---

`{button ,AL("VENTURA_BitmapSharpen;;;;','0,"Defaultoverview"),}` [Related Topics](#)

## .BitmapSharpenAdaptiveUnsharp (VENTURA)

**.BitmapSharpenAdaptiveUnsharp .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. Use the Adaptive Unsharp filter to sharpen edge detail in your bitmap.

Syntax	Description
<b>.Percentage</b>	Specifies the intensity of the effect. Valid values range from 0 to 100 (%).

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapSharpenAdaptiveUnsharp 70
```

---

{button ,AL('VENTURA\_BitmapSharpen;;;;';0,"Defaultoverview",)} [Related Topics](#)

## **.BitmapSharpenDirectionalSharpen (VENTURA)**

**.BitmapSharpenDirectionalSharpen .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. The Directional Sharpen filter enhances edge detail in your bitmap. Bitmap pixels of similar shades determine the direction in which the sharpening is applied.

---

### **Syntax**

### **Description**

**.Percentage**

Specifies the intensity of the effect. Valid values range from 0 to 100 (%).

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

`.BitmapSharpenDirectionalSharpen 10`

---

{button ,AL('VENTURA\_BitmapSharpen;;;;;'0,"Defaultoverview",)} [Related Topics](#)



# .BitmapSharpenFindEdges (VENTURA)

**.BitmapSharpenFindEdges .Level=*long*, .EdgeType=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Find Edges filter to convert outlines to soft or solid lines. For the clearest results, use the Find Edges filter on high-contrast bitmaps; for example, bitmaps that contain text.

Syntax	Description
.Level	Specifies the intensity of the effect. Valid values range from 0 to 100%.
.EdgeType	Specifies the type of edge: 0      Soft, blurry outline 1      Sharp, crisp outline

**Note**

-  This command was introduced in Corel VENTURA 8.
-  Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

**Example**

`.BitmapSharpenFindEdges 95, 1`

## **.BitmapSharpenHighPass (VENTURA)**

**.BitmapSharpenHighPass .Radius=*long*, .Percentage=*long***

This command applies a bitmap effect to a selected bitmap. The High Pass filter removes bitmap detail by emphasizing the highlights and luminous areas of your bitmap. Most bitmap detail is removed on high percentage settings.

Syntax	Description
<b>.Radius</b>	Specifies the range of the effect. Larger values increase the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
<b>.Percentage</b>	Specifies the intensity of the effect. Larger values remove more shadow detail. Valid values range from 0 to 100%.

### **Note**



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### **Example**

```
.BitmapSharpenHighPass 15, 57
```

---

{button ,AL('VENTURA\_BitmapSharpen;;;;','0',"Defaultoverview",)} [Related Topics](#)

## .BitmapSharpenUnsharpMask (VENTURA)

**.BitmapSharpenUnsharpMask .Radius=*long*, .Percentage=*long*, .Threshold=*long***

This command applies a bitmap effect to a selected bitmap. You can use the Unsharp Mask filter to sharpen the focus of a bitmap by increasing its edge detail.

Syntax	Description
<b>.Radius</b>	Specifies the range of the effect. Larger values increase the number of pixels that are successively selected and evaluated when you apply the effect. Valid values range from 1 to 20.
<b>.Percentage</b>	Specifies the intensity of the effect. Valid values range from 0 to 100(%).
<b>.Threshold</b>	Specifies the change in value that is required by a pixel before the Unsharp Mask effect is applied.

### Note



This command was introduced in Corel VENTURA 8.



Bitmaps must be embedded before you can manipulate them. In Corel VENTURA, you cannot manipulate an externally referenced bitmap or a bitmap inserted as an OLE object.

### Example

```
.BitmapSharpenUnsharpMask 10, 388, 42
```

---

{button ,AL("VENTURA\_BitmapSharpen;;;;";'0,"Defaultoverview",)} [Related Topics](#)

## **.BorderTagAddNew (VENTURA)**

**.BorderTagAddNew .TagName=*string*, .CopyFrom=*string***

This command is replaced with **RuleTagAddNew** in Corel VENTURA 8.

This command adds a new rule (border) tag to the current publication. The new tag is based on an existing rule (border) tag.

Syntax	Description
<b>.TagName</b>	Specifies the name of the new rule (border) tag. The tag name must not already exist in the style sheet.
<b>.CopyFrom</b>	Specifies the rule (border) tag that the new rule (border) tag is based on.

### **Note**



This command corresponds to the Tag Window. Click Tools, Tag Window.

### **Example**

```
.BorderTagAddNew "2P_Special", "2P_SOLID"
```

The above example adds a rule (border) tag named 2P\_Special to the current publication. The new tag is based on the 2P\_SOLID tag.

---

{button ,AL("VENTURA\_rules;;;','0,"Defaultoverview",,)} **Related Topics**

## **.BorderTagCopy (VENTURA)**

**.BorderTagCopy** .TagName=*string*

This command is replaced with **RuleTagCopy** in Corel VENTURA 8.

This command copies a rule (border) tag from the current style sheet to the Clipboard. This command can be used to copy tags into VENTURA Libraries.

Syntax	Description
.TagName	Specifies the name of the rule (border) tag to copy.

### **Note**



This command corresponds to the Tag Window. Click Tools, Tag Window.

### **Example**

```
.BorderTagCopy "TradeMark"  
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste
```

The above example copies the rule (border) tag TradeMark, creates a VENTURA library named MYLIB.VLB, and pastes the TradeMark tag into the new library.

---

{button ,AL("VENTURA\_rules;;;','0,"Defaultoverview",)} **Related Topics**



## .BorderTagCount (VENTURA)

**ReturnValue** & = .BorderTagCount ( )

This function is replaced with [RuleTagCount](#) in Corel VENTURA 8.

This function returns the number of rule (border) tags in the current style sheet.

Syntax	Description
<b>ReturnValue</b> &	Specifies the numeric variable that is assigned the number of rule (border) tags in the current style sheet.

### Note



This command cannot be recorded.

### Example

```
Bcounts = .BorderTagCount ( )
```

The number of rule (border) tags is assigned the **Bcounts** variable.

---

{button ,AL("VENTURA\_rules;;;','0,"Defaultoverview",,)} [Related Topics](#)

## **.BorderTagDelete (VENTURA)**

**.BorderTagDelete .TagName=*string*, .ReformatTagName=*string***

This command is replaced with [RuleTagDelete](#) in Corel VENTURA 8.

This command deletes a rule (border) tag from the current style sheet. Any item in the publication — and in other publications that are linked to the same style sheet through the VENTURA Library — formatted with the deleted tag will be reformatted using another tag of your choice.

Syntax	Description
<b>.TagName</b>	Specifies the name of the rule (border) tag to delete.
<b>.ReformatTagName</b>	Specifies the rule (border) tag to apply to borders tagged with <b>.TagName</b> .

### **Note**



This command corresponds to the Tag Window. Click Tools, Tag Window.



You cannot delete any generated tags (those with a Z\_ prefix).

### **Example**

```
.BorderTagDelete "Montreal", "Boston"
```

The above example deletes the rule (border) tag Montreal and applies the Boston tag to rules using the Montreal tag.

---

**{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} [Related Topics](#)**

## .BorderTagGetAt (VENTURA)

**ReturnString\$ = .BorderTagGetAt .TagIndex=*long***

This function is replaced with [RuleTagGetAt](#) in Corel VENTURA 8.

This function returns the rule (border) tag name associated with a rule (border) tag index number in the current style sheet.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of a rule (border) tag.
<b>.TagIndex</b>	Specifies the index number of a rule (border) tag. Index numbers are not associated with tag order in the Manage Tag List dialog box (alphabetical order). Instead, index numbers are associated by the order in which the rule (border) tags are created. The first tag created is 1, the second tag created is 2, and so on. The last created tag is equal to the return value from the <a href="#">.BorderTagCount</a> function. If a tag is deleted, the tag index is recompiled.

### Note



This command cannot be recorded.

### Example

```
LastTag& = .BorderTagCount ( )      'number of tags  
LastTagName$ = .BorderTagGetAt (LastTag&)
```

In the above example, the first statement counts the number of rule (border) tags. The second statement is assigned the name of last created rule (border) tag still available in the current style sheet.

---

{button ,AL("VENTURA\_rules;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.BorderTagRename (VENTURA)**


**.BorderTagRename .OldTagName=*string*, .NewTagName=*string***

This command is replaced with [RuleTagRename](#) in Corel VENTURA 8.

This command renames a specified rule (border) tag. Renaming tags may affect the formatting of other publications that are linked to the same style sheet through the Library.

<b>Syntax</b>	<b>Description</b>
<b>.OldTagName</b>	Specifies the rule (border) tag to be renamed.
<b>.NewTagName</b>	Specifies the new rule (border) tag name.

### **Note**

 This command corresponds to the Tag Window. Click Tools, Tag Window.

### **Example**

`.BorderTagRename "Name 1", "Name 2"`

The above example renames the rule (border) tag Name 1 to Name 2.

---

`{button ,AL('VENTURA_rules;;;','0',"Defaultoverview",)} Related Topics`

## **.FormatBorder (VENTURA)**


**.FormatBorder** .SpaceBefore1=*long*, .SpaceBefore2=*long*, .SpaceBefore3=*long*, .SpaceAfter3=*long*, .Overprint=*long*

In Corel VENTURA 8, this command has been replaced by [FormatRuleTag](#).

This command changes the spacing and overprint attributes of a rule (border). To change a rule (border) tag's attributes, this command must be enclosed by the [.FormatBorderTagBegin](#) and [.FormatBorderTagEnd](#) commands.

Syntax	Description
.SpaceBefore1	Specifies the spacing above a border's first ruling line in tenths of a micron. A border can be made up of three ruling lines.
.SpaceBefore2	Specifies the spacing above a border's tag's second ruling line in tenths of a micron. A border can be made up of three ruling lines.
.SpaceBefore3	Specifies the spacing above a border's tag's third ruling line in tenths of a micron. A border can be made up of three ruling lines.
.SpaceAfter3	Specifies the spacing below a border's tag's third ruling line in tenths of a micron. A border can be made up of three ruling lines.
.Overprint	Specifies whether to enable the overprint option. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0) which is the default if omitted. Enabling the overprint option allows the border's fill to print over the background color. This option is relevant only when printing color separations.

### **Note**

 This command corresponds to the Border Tag Style dialog box in Corel VENTURA. Click Format, Manage Tag List, Border. Select a tag and click Edit Tag.

### **Example**

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatBorderTagBegin "Sample"
    .FormatBorder 1*M_POINT, 2*M_POINT, 3*M_POINT, 4*M_POINT, TRUE
.FormatBorderTagEnd
```

The above example sets the Sample rule (border) tag's spacing attributes and enables the Overprint option.

The [LENGTHCONVERT](#) function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point.

---


{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} [Related Topics](#)

## .FormatBorderColor (VENTURA)


**.FormatBorderColor** *.LineIndex=long*, *.ColorModel=long*, *.Color1=long*, *.Color2=long*, *.Color3=long*, *.Color4=long*

In Corel VENTURA 8, this command has been replaced by **FormatRuleTagColor**.

This command changes the color attributes of a rule (border). To change a rule (border) tag's attributes, this command must be enclosed by the **.FormatBorderTagBegin** and **.FormatBorderTagEnd** commands.

Syntax	Description
<b>.LineIndex</b>	Specifies which ruling line in the border to set color attributes for. A border can be made of three ruling lines: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.ColorModel</b>	Specifies the color model to use for the specified ruling line: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>  If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note

 This command corresponds to the Fill Color dialog box in Corel VENTURA. Click Format, Manage Tag List, Border. Select a tag and click Edit Tag. Click the color swatch corresponding to the ruling line to edit, then click Others.

### Example

```
.FormatBorderTagBegin "Sample"  
  .FormatBorderColor 2, 5, 200, 25, 15  
.FormatBorderTagEnd
```

The above example sets the Sample rule (border) tag's second ruling line to red using the RGB color model.

---

{button ,AL('VENTURA\_rules;;;','0,"Defaultoverview",')} [Related Topics](#)

## .FormatBorderColorGet (VENTURA)

**.FormatBorderColorGet** *.LineIndex=long*, *.ColorModel=long*, *.Color1=long*, *.Color2=long*, *.Color3=long*, *.Color4=long*

In Corel VENTURA 8, this function has been replaced by [FormatRuleTagColorGet](#).

This function returns the color attributes of a specified rule. To specify the rule (border) tag, enclose the **.FormatBorderColorGet** function with the [.FormatBorderTagBegin](#) and [.FormatBorderTagEnd](#) commands.

Syntax	Description
<b>.LineIndex</b>	Specifies which ruling line in the border to return color attributes for. A border can be made of three ruling lines: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.ColorModel</b>	Specifies the numeric variable that is assigned a return value corresponding to the specified ruling line's uniform color model: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color3</b>	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color4</b>	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

### Note

**I** The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example

```
.FormatBorderTagBegin "Sample"  
  .FormatBorderColorGet 2, cm&, c1&, c2&, c3&, c4&  
.FormatBorderTagEnd
```

In the above example the **.FormatBorderColorGet** function returns the Sample rule (border) tag's second ruling line color attributes to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

---

{button ,AL('VENTURA\_rules;;;;',0,"Defaultoverview"),} [Related Topics](#)

# **.FormatBorderGet (VENTURA)**

**.FormatBorderGet** *.SpaceBefore1=long, .SpaceBefore2=long, .SpaceBefore3=long, .SpaceAfter3=long, .Overprint=long*

In Corel VENTURA 8, this function has been replaced by **FormatRuleTagGet**.

This function returns the spacing and overprint attributes of a specified rule. To specify the rule (border) tag, enclose the **.FormatBorderGet** function with the **.FormatBorderTagBegin** and **.FormatBorderTagEnd** commands.

Syntax	Description
.SpaceBefore1	Specifies a numeric variable that is assigned the space above a border's first ruling line in tenths of a micron. A border can be made up of three ruling lines.
.SpaceBefore2	Specifies a numeric variable that is assigned the space above a border's second ruling line in tenths of a micron. A border can be made up of three ruling lines.
.SpaceBefore3	Specifies a numeric variable that is assigned the space above a border's third ruling line in tenths of a micron. A border can be made up of three ruling lines.
.SpaceAfter3	Specifies a numeric variable that is assigned the space below a border's third ruling line in tenths of a micron. A border can be made up of three ruling lines.
.Overprint	Specifies a variable that is assigned a value corresponding to whether a border has its overprint option enabled: TRUE (-1) if the option is enabled; otherwise, FALSE (0).

## **Note**

**I** The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

## **Example**

```
DIM OP AS BOOLEAN
.FormatBorderTagBegin "Sample"
    .FormatBorderGet SB1&, SB2&, SB3&, SA&, OP
.FormatBorderTagEnd
```

In above example the **.FormatBorderGet** function returns the spacing and overprint attributes for the Sample rule (border) tag's variables **SB1**, **SB2**, **SB3**, **SA**, and **OP**.



## **.FormatBorderOutline (VENTURA)**

**.FormatBorderOutline** *.LineIndex=long*, *.Width=long*, *.Type=long*, *.EndCaps=long*, *.JoinType=long*, *.Aspect=long*, *.Angle=long*, *.DotDash=long*, *.RightArrow=long*, *.LeftArrow=long*, *.BehindFill=Boolean*

In Corel VENTURA 8, this command has been replaced by **FormatRuleTagColorGet**.

This command changes the outline attributes of a rule. To change a rule (border) tag's attributes, this command must be enclosed by the **.FormatBorderTagBegin** and **.FormatBorderTagEnd** commands.

Syntax	Description
<b>.LineIndex</b>	Specifies which ruling line in the border to set attributes for. A border can be made of three ruling lines: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.Width</b>	Specifies the width of the specified ruling line in tenths of a micron.
<b>.Type</b>	Specifies the style of the specified ruling line: 0 None 1 Solid 2 Dotted or Dashed  If this parameter is set to Dotted or Dashed, the <b>.DotDash</b> parameter determines the style.
<b>.EndCaps</b>	Specifies the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the width of the nib. A square nib has a value of 100; smaller values make the nib narrower, creating a calligraphy effect. Valid values range from 1 to 100, inclusive.
<b>.Angle</b>	Specifies the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree). This option creates a calligraphic effect.
<b>.DotDash</b>	Specifies the style of the specified ruling line if the <b>.Type</b> parameter is set to Dotted or Dashed. The styles correspond to the Style list box in Corel VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the style of the right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right) — the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the style of the left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right) — the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies whether the specified ruling line is placed behind, or in front of, the object's fill. When placed behind, only half the specified ruling line's thickness is visible. This option is particularly useful for outlined text. Set to TRUE (-1) to enable the behind fill option; otherwise, set to FALSE (0), which is the default if omitted.

### **Note**

**1** This command corresponds to Corel VENTURA's Outline Pen dialog box. Click Format, Manage Tag List, Border. Select a tag and click Edit Tag. Click the pen icon corresponding to the ruling line to edit.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatBorderTagBegin "Sample"
.FormatBorderOutline 2, 0.05*M_INCH, 2, 0, 0, 90, 25, 7
.FormatBorderTagEnd
```

The above example sets the Sample rule (border) tag's second ruling line to 0.05 inches with a dashed line.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL(^VENTURA\_rules;;;',0,"Defaultoverview",)} Related Topics

## **.FormatBorderOutlineGet (VENTURA)**

**.FormatBorderOutlineGet** *.LineIndex=long*, *.Width=long*, *.Type=long*, *.EndCaps=long*, *.JoinType=long*, *.Aspect=long*, *.Angle=long*, *.DotDash=long*, *.RightArrow=long*, *.LeftArrow=long*, *.BehindFill=Boolean*

In Corel VENTURA 8, this function has been replaced by [FormatRuleTagOutlineGet](#).

This function returns the outline attributes of a specified rule. To specify the rule (border) tag, enclose the **.FormatBorderOutlineGet** function with the [.FormatBorderTagBegin](#) and [.FormatBorderTagEnd](#) commands.

Syntax	Description
<b>.LineIndex</b>	Specifies which ruling line in the border to return attributes for. A border can be made of three ruling lines: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.Width</b>	Specifies the numeric variable that is assigned the width of the specified ruling line in tenths of a micron.
<b>.Type</b>	Specifies the numeric variable that is assigned the style of the specified ruling line: 0 None 1 Solid 2 Dotted or Dashed If this parameter assigns Dotted or Dashed, the <b>.DotDash</b> parameter assigns the style.
<b>.EndCaps</b>	Specifies the numeric variable that is assigned the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the numeric variable that is assigned the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the numeric variable that is assigned the width of the nib.
<b>.Angle</b>	Specifies the numeric variable that is assigned the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree).
<b>.DotDash</b>	Specifies the numeric variable that is assigned the style of the specified ruling line if the <b>.Type</b> parameter is set to Dotted or Dashed. The styles correspond to the Style list box in Corel VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the numeric variable that is assigned the style of the right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right) — the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the numeric variable that is assigned the style of the left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right) — the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies the numeric variable that is assigned a value indicating whether the Behind Fill option is enabled: TRUE (-1) if enabled; otherwise, FALSE (0).

### **Note**



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### **Example**

```
.FormatBorderTagBegin "Sample"  
    .FormatBorderOutlineGet 2, Owidth&, Otype&, , , , Odotdash&  
.FormatBorderTagEnd
```

In above example the **.FormatBorderOutlineGet** function returns the outline attributes for the Sample rule (border) tag's second ruling line to the variables **Owidth**, **Otype**, and **Odotdash**.

## **.FormatBorderTagBegin (VENTURA)**

**.FormatBorderTagBegin** *.TagName=string*

In Corel VENTURA 8, this command has been replaced by [FormatRuleTagBegin](#).

The **.FormatBorderTagBegin** and [FormatBorderTagEnd](#) commands are required to change a rule (border) tag's attributes. The two commands enclose Corel VENTURA commands that can modify a rule (border) tag. The **.FormatBorderTagBegin** statement is placed before the first rule (border) modifying command and the **FormatBorderTagEnd** command is placed after the last rule (border) modifying command.

Syntax	Description
--------	-------------

<b>.TagName</b>	Specifies the rule (border) tag to modify. The tag must already exist in the publication.
-----------------	---

### **Note**

The following commands can be used to modify a rule (border) tag.



[FormatBorder command](#)



[FormatBorderColor command](#)



[FormatBorderOutline command](#)

### **Example**

```
.FormatBorderTagBegin "NewTag"
    .FormatBorder .....
    .FormatBorderColor .....
    .FormatBorderOutline .....
.FormatBorderTagEnd
```

---

{button ,AL('VENTURA\_rules;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.FormatBorderTagEnd (VENTURA)**

### **.FormatBorderTagEnd**

In Corel VENTURA 8, this command has been replaced by **FormatRuleTagEnd**.

The **.FormatBorderTagBegin** and **.FormatBorderTagEnd** commands are required to change a rule (border) tag's attributes. The two commands enclose Corel VENTURA commands that can modify a rule (border) tag. The **.FormatBorderTagBegin** statement is placed before the first border modifying command and the **.FormatBorderTagEnd** command is placed after the last rule (border) modifying command.

#### **Note**

The following commands can be used to modify a rule (border) tag.



**FormatBorder command**



**FormatBorderColor command**



**FormatBorderOutline command**

#### **Example**

```
.FormatBorderTagBegin "NewTag"
    .FormatBorder .....
    .FormatBorderColor .....
    .FormatBorderOutline .....
.FormatBorderTagEnd
```

---

{button ,AL('VENTURA\_rules;;;',0,"Defaultoverview"),} **Related Topics**

## .FormatRule (VENTURA)

**.FormatRule** **.RuleIndex=long**, **.LineIndex=long**, **.Thickness=long**, **.SpaceBefore=long**, **.SpaceAfter=long**, **.WidthStyle=long**, **.CustomWidth=long**, **.CustomIndent=long**

This command modifies a specified rule for a paragraph or frame. A rule can be made up of three lines. This command should be enclosed in the **FormatObjectBegin** and **FormatObjectEnd** construct and should follow the **FormatRuleStyle** command (see the example below).



The following commands can be used to modify rules for a paragraph or frame:

-  **FormatRule**
-  **FormatRuleStyle**
-  **FormatRuleSide**
-  **FormatRuleColor**
-  **FormatRuleOutline**

See **FormatRuleTagBegin** for information about using rule tags in Corel SCRIPT.

Syntax	Description
<b>.RuleIndex</b>	Specifies the rule to modify: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right  This rule must already exist. Rules are created with the <b><u>FormatRuleStyle</u></b> command.
<b>.LineIndex</b>	Specifies the rule's line to be modified: 1 First line 2 Second line 3 Third line
<b>.Thickness</b>	Specifies the thickness of the specified line in tenths of a micron.
<b>.SpaceBefore</b>	Specifies the space before the specified line in tenths of a micron.
<b>.SpaceAfter</b>	Specifies the space after the third line in tenths of a micron. This parameter should not be used if <b>.LineIndex</b> is set to 1 or 2.
<b>.WidthStyle</b>	Specifies the rule width: 0 Paragraph 1 Margin 2 Column 3 Frame 4 Custom
<b>.CustomWidth</b>	If <b>.WidthStyle</b> is set to Custom, specifies the width of the specified rule in tenths of a micron.
<b>.CustomIndent</b>	If <b>.WidthStyle</b> is set to Custom, specifies the left indent for the current rule in tenths of a micron.

### Note

-  This command was introduced in Corel VENTURA 8.
-  This command corresponds to the Rules tab on the Frame Properties or Paragraph properties dialog box.

### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatObjectBegin 2
.FormatRuleStyle 3
.FormatRuleSide 3, 4, 0, -1.25*M_POINT
.FormatRule 1, 1, .25*M_POINT, 0, 0, 3
.FormatRuleColor 1, 1, 2, 0, 40, 60, 0
.FormatRule 2, 1, .25*M_POINT, 0, 0, 3
.FormatRuleColor 2, 1, 2, 0, 40, 60, 0
.FormatRuleOutline 2, 1, , 2, 0, 0, 90, 25, 7
.FormatObjectEnd
```

'format selected frame  
'rule above and below  
'modifies the Above line  
'modifies the Above line  
'modifies the Below line  
'modifies the Below line

In the above example, the **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a

micron in a point. The **FormatObjectBegin** command specified that a selected frame will be modified.

After the **FormatObjectBegin** command, the first line specifies to apply rules above and below ( $1 + 2 = 3$ ) the active frame. The second line modifies the indentation of the rule. The third line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The fourth line modifies the Above line's first line color setting using the CMYK model. The fifth and sixth lines modify the Below line the same way the Above lines were modified. The seventh line modifies the Below line by making it dashed.

---

`{button ,AL("VENTURA_rules;;;",0,"Defaultoverview",)} Related Topics`

## .FormatRuleColor (VENTURA)


**.FormatRuleColor** **.RuleIndex=long**, **.LineIndex=long**, **.ColorModel=long**, **.Color1=long**, **.Color2=long**, **.Color3=long**, **.Color4=long**

This command modifies a specified rule's color settings for a paragraph or frame. A rule can be made up of three lines. This command should be enclosed in the **FormatObjectBegin** and **FormatObjectEnd** construct and should follow the **FormatRuleStyle** command (see the example below).



The following commands can be used to modify rules for a paragraph or frame:

-  **FormatRule**
-  **FormatRuleStyle**
-  **FormatRuleSide**
-  **FormatRuleColor**
-  **FormatRuleOutline**

See **FormatRuleTagBegin** for information about using rule tags in Corel SCRIPT.

Syntax	Description
<b>.RuleIndex</b>	Specifies the rule to modify: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right  This rule must already exist. Rules are created with the <b>FormatRuleStyle</b> command.
<b>.LineIndex</b>	Specifies the rule's line to be modified: 1 First line 2 Second line 3 Third line
<b>.ColorModel</b>	Specifies the color model to use for the specified line: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u>  If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.
<b>.Color1</b>	Specifies the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.
<b>.Color3</b>	Specifies the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.
<b>.Color4</b>	Specifies the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.

### Note

-  This command was introduced in Corel VENTURA 8.
-  This command corresponds to the Rules tab on the Frame Properties or Paragraph properties dialog box.



### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatObjectBegin 2                                'format selected frame
.FormatRuleStyle 3                                  'rule above and below
.FormatRuleSide 3, 4, 0, -1.25*M_POINT
.FormatRule 1, 1, .25*M_POINT, 0, 0, 3             'modifies the Above line
.FormatRuleColor 1, 1, 2, 0, 40, 60, 0             'modifies the Above line
.FormatRule 2, 1, .25*M_POINT, 0, 0, 3             'modifies the Below line
.FormatRuleColor 2, 1, 2, 0, 40, 60, 0             'modifies the Below line
.FormatRuleOutline 2, 1, , 2, 0, 0, 90, 25, 7
.FormatObjectEnd
```

In the above example, the LENGTHCONVERT function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point. The FormatObjectBegin command specified that a selected frame will be modified.

After the **FormatObjectBegin** command, the first line specifies to apply rules above and below ( $1 + 2 = 3$ ) the active frame. The second line modifies the indentation of the rule. The third line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The fourth line modifies the Above line's first line color setting using the CMYK model. The fifth and sixth lines modify the Below line the same way the Above lines were modified. The seventh line modifies the Below line by making it dashed.

---

{button ,AL('VENTURA\_rules;;;;','0,"Defaultoverview",)} Related Topics

## .FormatRuleColorGet (VENTURA)

**.FormatRuleColorGet .RuleIndex=long, .LineIndex=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long**

This function returns the color attributes of a specified rule of the active paragraph or frame. A rule can be made up of three lines.

Syntax	Description
<b>.RuleIndex</b>	Specifies the rule to return attributes for: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right
<b>.LineIndex</b>	Specifies the line to return color attributes for. An rule can be made of 3 lines: 1 First line 2 Second line 3 Third line
<b>.ColorModel</b>	Specifies the numeric variable that is assigned a return value corresponding to the specified line's uniform color model: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u>
<b>.Color1</b>	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color3</b>	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color4</b>	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

### Note



This function was introduced in Corel VENTURA 8.



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatRuleColorGet 2, 2, cm&, c1&, c2&, c3&, c4&
```

In the above example the **.FormatRuleColorGet** function returns the color attributes for the second line in the Below rule to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

---

{button ,AL('VENTURA\_rules;;;',0,"Defaultoverview"),} [Related Topics](#)

## .FormatRuleGet (VENTURA)

**.FormatRuleGet** *.RuleIndex=long*, *.LineIndex=long*, *.Thickness=long*, *.SpaceBefore=long*, *.SpaceAfter=long*, *.WidthStyle=long*, *.CustomWidth=long*, *.CustomIndent=long*

This function returns the attributes of a specified line of a rule of the active paragraph or frame. A rule can be made up of three lines.

Syntax	Description
<b>.RuleIndex</b>	Specifies the rule to return attributes for: 0     None 1     Above 2     Below 4     Box 8     Left 16    Right
<b>.LineIndex</b>	Specifies the line to return color attributes for. An rule can be made up of three lines: 1     First line 2     Second line 3     Third line
<b>.Thickness</b>	Specifies the numeric variable that is assigned the thickness of the specified line in tenths of a micron.
<b>.SpaceBefore</b>	Specifies the numeric variable that is assigned the space before the line in tenths of a micron.
<b>.SpaceAfter</b>	Specifies the numeric variable that is assigned the space after the third line in tenths of a micron. This parameter should not be used if <b>.LineIndex</b> is set to 1 or 2.
<b>.WidthStyle</b>	Specifies the numeric variable that is assigned the rule width: 0     Paragraph 1     Margin 2     Column 3     Frame 4     Custom
<b>.CustomWidth</b>	If <b>.WidthStyle</b> is set to Custom, specifies the numeric variable that is assigned the width of the specified rule in tenths of a micron.
<b>.CustomIndent</b>	If <b>.WidthStyle</b> is set to Custom, specifies the numeric variable that is assigned the left indent for the current rule in tenths of a micron.

### Note



- This function was introduced in Corel VENTURA 8.
- The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
- This command cannot be recorded.

### Example

```
.FormatRuleGet 2, 2, Thick1&, SpaceBefore&
```

In the above example, the **.FormatRuleGet** function returns the attributes for the second line in the Below rule to the variables **Thick1** and **SpaceBefore**.

---

{button ,AL('VENTURA\_rules;;;;',0,"Defaultoverview",)} Related Topics

## **.FormatRuleOutline (VENTURA)**


**.FormatRuleOutline .RuleIndex=*long*, .LineIndex=*long*, .Width=*long*, .Type=*long*, .EndCaps=*long*, .JoinType=*long*, .Aspect=*long*, .Angle=*long*, .DotDash=*long*, .RightArrow=*long*, .LeftArrow=*long*, .BehindFill=*Boolean***

This command modifies a specified rule's outline setting for a paragraph or frame. A rule can be made up of three lines. This command should be enclosed in the **FormatObjectBegin** and **FormatObjectEnd** construct and should follow the **FormatRuleStyle** command (see the example below).

The following commands can be used to modify rules for a paragraph or frame:

-  **FormatRule**
-  **FormatRuleStyle**
-  **FormatRuleSide**
-  **FormatRuleColor**
-  **FormatRuleOutline**

See **FormatRuleTagBegin** for information about using rule tags in Corel SCRIPT.

Syntax	Description
<b>.RuleIndex</b>	Specifies the rule to modify: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right  This rule must already exist. Rules are created with the <b><u>FormatRuleStyle</u></b> command.
<b>.LineIndex</b>	Specifies the rule's line to be modified: 1 First line 2 Second line 3 Third line
<b>.Width</b>	Specifies the thickness of the specified line in tenths of a micron.
<b>.Type</b>	Specifies the style of the specified line: 0 None 1 Solid 2 Dotted or Dashed  If this parameter is set to Dotted or Dashed, the <b>.DotDash</b> parameter determines the style.
<b>.EndCaps</b>	Specifies the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the line corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the width of the nib. A square nib has a value of 100; smaller values make the nib narrower, creating a calligraphy effect. Valid values range from 1 to 100, inclusive.
<b>.Angle</b>	Specifies the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree). This option creates a calligraphic effect.
<b>.DotDash</b>	Specifies the style of the specified rule if <b>.Type</b> is set to Dotted or Dashed. The styles correspond to the Style list box in Corel VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the style of the right arrow. Right-arrow styles are listed in the right arrow list box of Corel VENTURA's Outline Pen dialog box. The styles are numbered and identified according to their position in the list (from left to right) — the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the style of the left arrow. Left-arrow styles are listed in the left arrow list box of Corel VENTURA's Outline Pen dialog box. The styles are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies whether the specified line is placed behind, or in front of, the object's fill. When

placed behind, only half the specified line's thickness is visible. This option is particularly useful for outlined text. Set to TRUE (-1) to enable the Behind Fill option; otherwise, set to FALSE (0), which is the default if omitted.

#### Note



This command was introduced in Corel VENTURA 8.



This command corresponds to the Rules tab on the Frame Properties or Paragraph properties dialog box.

#### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatObjectBegin 2                                'format selected frame
.FormatRuleStyle 3                                  'rule above and below
.FormatRuleSide 3, 4, 0, -1.25*M_POINT
.FormatRule 1, 1, .25*M_POINT, 0, 0, 3              'modifies the Above line
.FormatRuleColor 1, 1, 2, 0, 40, 60, 0              'modifies the Above line
.FormatRule 2, 1, .25*M_POINT, 0, 0, 3              'modifies the Below line
.FormatRuleColor 2, 1, 2, 0, 40, 60, 0              'modifies the Below line
.FormatRuleOutline 2, 1, , 2, 0, 0, 90, 25, 7
.FormatObjectEnd
```

In the above example, the LENGTHCONVERT function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point. The FormatObjectBegin command specified that a selected frame will be modified.

After the **FormatObjectBegin** command, the first line specifies to apply rules above and below (1 + 2 = 3) the active frame. The second line modifies the indentation of the rule. The third line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The fourth line modifies the Above line's first line color setting using the CMYK model. The fifth and sixth lines modify the Below line the same way the Above lines were modified. The seventh line modifies the Below line by making it dashed.



---

{button ,AL('VENTURA\_rules;;;',0,"Defaultoverview"),} Related Topics



## .FormatRuleOutlineGet (VENTURA)

**.FormatRuleOutlineGet** *.RuleIndex=long, .LineIndex=long, .Width=long, .Type=long, .EndCaps=long, .JoinType=long, .Aspect=long, .Angle=long, .DotDash=long, .RightArrow=long, .LeftArrow=long, .BehindFill=Boolean*

This function returns the pen settings for a specified rule of the active paragraph or frame. A rule can be made up of three lines.

Syntax	Description
<b>.RuleIndex</b>	Specifies the rule to return attributes for: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right
<b>.LineIndex</b>	Specifies the line to return attributes for. An rule can be made up of three lines: 1 First line 2 Second line 3 Third line
<b>.Width</b>	Specifies the numeric variable that is assigned the width of the specified line in tenths of a micron.
<b>.Type</b>	Specifies the numeric variable that is assigned the style of the specified line: 0 None 1 Solid 2 Dotted or Dashed If this parameter assigns Dotted or Dashed, the <b>.DotDash</b> parameter assigns the specific style.
<b>.EndCaps</b>	Specifies the numeric variable that is assigned the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the numeric variable that is assigned the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the numeric variable that is assigned the width of the nib.
<b>.Angle</b>	Specifies the numeric variable that is assigned the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree).
<b>.DotDash</b>	Specifies the numeric variable that is assigned the style of the specified rule if <b>.Type</b> is set to Dotted or Dashed. The styles correspond to the Style list box in Core! VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the numeric variable that is assigned the style of the right arrow. Right-arrow styles are listed in the right arrow list box of VENTURA's Outline Pen dialog box. The styles are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the numeric variable that is assigned the style of the left arrow. Left-arrow styles are listed in the left arrow list box of VENTURA's Outline Pen Roll-Up. The styles are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies the numeric variable that is assigned a value indicating whether the Behind Fill option is enabled. If the option is enabled, TRUE (-1) is assigned; otherwise, FALSE (0).

### Note

-  This function was introduced in Core! VENTURA 8.
-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example

```
.FormatRuleOutlineGet 2, 2, Owidth&, Otype&, , , , Odotdash&
```

In the above example the **.FormatRuleOutlineGet** function returns the attributes for the second line in the Below rule to the variables **Owidth**, **Otype**, and **Odotdash**.

---

{button ,AL(^VENTURA\_rules;;;',0,"Defaultoverview",)} Related Topics

## .FormatRuleSide (VENTURA)

**.FormatRuleSide** *.RuleIndex=long, .WidthStyle=long, .CustomWidth=long, .CustomIndent=long, .RuleName=string*

This command modifies a specified rule's width and tag setting for a paragraph or frame. This command should be enclosed in the **FormatObjectBegin** and **FormatObjectEnd** construct and should follow the **FormatRuleStyle** command (see the example below).



The following commands can be used to modify rules for a paragraph or frame:

-  **FormatRule**
-  **FormatRuleStyle**
-  **FormatRuleSide**
-  **FormatRuleColor**
-  **FormatRuleOutline**

See **FormatRuleTagBegin** for information about using rule tags in Corel SCRIPT.

Syntax	Description
<b>.RuleIndex</b>	Specifies the rule to modify: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right  This rule must already exist. Rules are created with the <b>FormatRuleStyle</b> command.
<b>.WidthStyle</b>	Specifies the rule width: 0 Paragraph 1 Margin 2 Column 3 Frame 4 Custom
<b>.CustomWidth</b>	If <b>.WidthStyle</b> is set to Custom, specifies the width of the specified rule in tenths of a micron.
<b>.CustomIndent</b>	If <b>.WidthStyle</b> is set to Custom, specifies the left indent for the current rule in tenths of a micron.
<b>.RuleName</b>	Specifies the rule tag to apply to the rule.

### Note

-  This command was introduced in Corel VENTURA 8.
-  This command corresponds to the Rules tab on the Frame Properties or Paragraph properties dialog box.

### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatObjectBegin 2                                'format selected frame
.FormatRuleStyle 3                                  'rule above and below
.FormatRuleSide 3, 4, 0, -1.25*M_POINT
.FormatRule 1, 1, .25*M_POINT, 0, 0, 3              'modifies the Above line
.FormatRuleColor 1, 1, 2, 0, 40, 60, 0              'modifies the Above line
.FormatRule 2, 1, .25*M_POINT, 0, 0, 3              'modifies the Below line
.FormatRuleColor 2, 1, 2, 0, 40, 60, 0              'modifies the Below line
.FormatRuleOutline 2, 1, , 2, 0, 0, 90, 25, 7
.FormatObjectEnd
```

In the above example, the **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point. The **FormatObjectBegin** command specified that a selected frame will be modified.

After the **FormatObjectBegin** command, the first line specifies to apply rules above and below ( $1 + 2 = 3$ ) the active frame. The second line modifies the indentation of the rule. The third line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The fourth line modifies the Above line's first line color setting using the CMYK model. The fifth and sixth lines modify the Below line the same way the Above lines were modified. The seventh line modifies the Below line by making it dashed.



## .FormatRuleSideGet (VENTURA)

**.FormatRuleSideGet** **.RuleIndex=long**, **.WidthStyle=long** , **.CustomWidth=long** , **.CustomIndent=long** , **.RuleName=string**

This function returns the attributes of a rule of the active paragraph or frame. A rule can be made up of three lines.

Syntax	Description
<b>.RuleIndex</b>	Specifies the rule to return attributes for: 0     None 1     Above 2     Below 4     Box 8     Left 16    Right
<b>.WidthStyle</b>	Specifies the numeric variable that is assigned the rule width: 0     Paragraph 1     Margin 2     Column 3     Frame 4     Custom
<b>.CustomWidth</b>	If <b>.WidthStyle</b> is set to Custom, specifies the numeric variable that is assigned the width of the specified rule in tenths of a micron.
<b>.CustomIndent</b>	If <b>.WidthStyle</b> is set to Custom, specifies the numeric variable that is assigned the left indent for the current rule in tenths of a micron.
<b>.RuleName</b>	If a rule has been applied to the specified rule, specifies a string variable that is assigned the rule name.

### Note



This function was introduced in Corel VENTURA 8.



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example

```
.FormatRuleSideGet 2, WRule&, CW&, CI&
```

In the above example, the **.FormatRuleSideGet** function returns the attributes for the Below rule to the variables **WRule**, **CW**, and **CI**.

---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} [Related Topics](#)

# .FormatRuleStyle (VENTURA)

.FormatRuleStyle .RuleStyle=*long*

This command specifies the rules to apply to the active paragraph or frame. A rule can be made up of three lines. This command should be enclosed in the [FormatObjectBegin](#) and [FormatObjectEnd](#) construct (see the example below).



The following commands can be used to modify rules for a paragraph or frame:

-  [FormatRule](#)
-  [FormatRuleStyle](#)
-  [FormatRuleSide](#)
-  [FormatRuleColor](#)
-  [FormatRuleOutline](#)

See [FormatRuleTagBegin](#) for information about using rule tags in Corel SCRIPT.

Syntax	Description
.RuleStyle	Specifies the rules to apply: <div><div>0None</div><div>1Above</div><div>2Below</div><div>4Box</div><div>8Left</div><div>16Right</div></div> <div>Rules can be combined by adding their corresponding values. For Above &amp; Below, set this parameter to 3.</div>

## Note

-  This command was introduced in Corel VENTURA 8.
-  This command corresponds to the Rules tab on the Frame Properties or Paragraph properties dialog box.

## Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatObjectBegin 2                                'format selected frame
.FormatRuleStyle 3                                  'rule above and below
.FormatRuleSide 3, 4, 0, -1.25*M_POINT
.FormatRule 1, 1, .25*M_POINT, 0, 0, 3              'modifies the Above line
.FormatRuleColor 1, 1, 2, 0, 40, 60, 0              'modifies the Above line
.FormatRule 2, 1, .25*M_POINT, 0, 0, 3              'modifies the Below line
.FormatRuleColor 2, 1, 2, 0, 40, 60, 0              'modifies the Below line
.FormatRuleOutline 2, 1, , 2, 0, 0, 90, 25, 7
.FormatObjectEnd
```

In the above example, the [LENGTHCONVERT](#) function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point. The [FormatObjectBegin](#) command specified that a selected frame will be modified.

After the [FormatObjectBegin](#) command, the first line specifies to apply rules above and below (1 + 2 = 3) the active frame. The second line modifies the indentation of the rule. The third line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The fourth line modifies the Above line's first line color setting using the CMYK model. The fifth and sixth lines modify the Below line the same way the Above lines were modified. The seventh line modifies the Below line by making it dashed.

## .FormatRuleStyleGet (VENTURA)

**.FormatRuleStyleGet** .RuleStyle=*long*

This function returns which rules are applied to the active paragraph or frame.

Syntax	Description
.RuleStyle	Specifies the numeric variable that is assigned the rules that are applied: 0     None 1     Above 2     Below 4     Box 8     Left 16    Right  Lines can be combined by adding their corresponding values. For example, if Above & Below are both enabled, this parameter returns 3.

### Note



This function was introduced in Corel VENTURA 8.



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatRuleStyleGet RuleStyle&
```

In the above example the function returns which rules are applied to the active paragraph or frame to the **RuleStyle** variable.

In cases where multiple lines are returned, you can use the AND (bitwise) operator to determine specific lines. The following example shows how to test if the variable used in the above example includes the Left line:

```
IF 8 AND RuleStyle& THEN LeftTest$ = "Yes" ELSE LeftTest$ = "No"
```

8 is the value of Left line. The variable **LeftTest** is assigned a string based on bitwise comparison. In this example, **LeftTest** is assigned **Yes** if **RuleStyle** has the Left line.

---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview",)} Related Topics

## .FormatRuleTag (VENTURA)

**.FormatRuleTag** *.SpaceBefore1=long, .SpaceBefore2=long, .SpaceBefore3=long, .SpaceAfter3=long, .Overprint=long*

This command changes the spacing and overprint attributes of a rule tag. To change a rule tag's attributes, this command must be enclosed by the **.FormatRuleTagBegin** and **.FormatRuleTagEnd** commands.

Syntax	Description
<b>.SpaceBefore1</b>	Specifies the spacing above a rule's first line in tenths of a micron. A rule can be made up of three lines.
<b>.SpaceBefore2</b>	Specifies the spacing above a rule's second line in tenths of a micron. A rule can be made up of three lines.
<b>.SpaceBefore3</b>	Specifies the spacing above a rule's third line in tenths of a micron. A rule can be made up of three lines.
<b>.SpaceAfter3</b>	Specifies the spacing below a rule's third line in tenths of a micron. A rule can be made up of three lines.
<b>.Overprint</b>	Specifies whether to enable the overprint option. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0) which is the default if omitted. Enabling the overprint option allows the rule's fill to print over the background color. This option is relevant only when printing color separations.

### Note



This command was introduced in Corel VENTURA 8.



This command corresponds to the Rule Tag Properties dialog box.

### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatRuleTagBegin "Sample"
.FormatRuleTag 1*M_POINT, 2*M_POINT, 3*M_POINT, 4*M_POINT, TRUE
.FormatRuleTagEnd
```

The above example sets the Sample rule tag's spacing attributes and enables the Overprint option.

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point.

---

{button ,AL('VENTURA\_rules;;;','0','Defaultoverview',)} [Related Topics](#)

## .FormatRuleTagBegin (VENTURA)

**.FormatRuleTagBegin** .TagName=*string*

The **.FormatRuleTagBegin** and **.FormatRuleTagEnd** commands are required to change a rule tag's attributes. The two commands enclose Corel VENTURA commands that can modify a rule tag. The **.FormatRuleTagBegin** statement is placed before the first rule modifying command and the **.FormatRuleTagEnd** command is placed after the last rule modifying command.

Syntax	Description
<b>.TagName</b>	Specifies the rule tag to modify. The tag must already exist in the publication.

### Note

The following commands can be used to modify a rule tag.



**FormatRuleTag command**



**FormatRuleTagColor command**



**FormatRuleTagOutline command**



This command was introduced in Corel VENTURA 8.

### Example

```
.FormatRuleTagBegin "NewTag"
    .FormatRuleTag .....
    .FormatRuleTagColor .....
    .FormatRuleTagOutline .....
.FormatRuleTagEnd
```


---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} [Related Topics](#)

## .FormatRuleTagColor (VENTURA)

**.FormatRuleTagColor** *.LineIndex=long*, *.ColorModel=long*, *.Color1=long*, *.Color2=long*, *.Color3=long*, *.Color4=long*

This command changes the color attributes of a rule tag. To change a rule tag's attributes, this command must be enclosed by the **.FormatRuleTagBegin** and **.FormatRuleTagEnd** commands.

Syntax	Description
<b>.LineIndex</b>	Specifies which line in the rule to set color attributes for. A rule can be made of three lines: 1 First line 2 Second line 3 Third line
<b>.ColorModel</b>	Specifies the color model to use for the specified line: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>  If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command was introduced in Corel VENTURA 8.



This command corresponds to the Rule Tag Properties dialog box.

### Example

```
.FormatRuleTagBegin "Sample"  
    .FormatRuleTagColor 2, 5, 200, 25, 15  
.FormatRuleTagEnd
```

The above example sets the Sample rule tag's second line to red using the RGB color model.

---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} [Related Topics](#)

## .FormatRuleTagColorGet (VENTURA)

**.FormatRuleTagColorGet** *.LineIndex=long*, *.ColorModel=long*, *.Color1=long*, *.Color2=long*, *.Color3=long*, *.Color4=long*

This function returns the color attributes of a specified rule tag. To specify the rule tag, enclose the **.FormatRuleTagColorGet** function with the **.FormatRuleTagBegin** and **.FormatRuleTagEnd** commands.

Syntax	Description
<b>.LineIndex</b>	Specifies which line in the rule to return color attributes for. A rule can be made of three lines: 1 First line 2 Second line 3 Third line
<b>.ColorModel</b>	Specifies the numeric variable that is assigned a return value corresponding to the specified line's uniform color model: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color3</b>	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color4</b>	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

### Note



This function was introduced in Corel VENTURA 8.



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example

```
.FormatRuleTagBegin "Sample"  
    .FormatRuleTagColorGet 2, cm&, c1&, c2&, c3&, c4&  
.FormatRuleTagEnd
```

In the above example the **.FormatRuleTagColorGet** function returns the Sample rule tag's second line color attributes to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

---

{button ,AL('VENTURA\_rules;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.FormatRuleTagEnd (VENTURA)**

### **.FormatRuleTagEnd**

The **.FormatRuleTagBegin** and **.FormatRuleTagEnd** commands are required to change a rule tag's attributes. The two commands enclose Corel VENTURA commands that can modify a rule tag. The **.FormatRuleTagBegin** statement is placed before the first rule modifying command and the **.FormatRuleTagEnd** command is placed after the last rule modifying command.

#### **Note**

The following commands can be used to modify a rule tag.



**FormatRuleTag command**



**FormatRuleTagColor command**



**FormatRuleTagOutline command**



This command was introduced in Corel VENTURA 8.

#### **Example**

```
.FormatRuleTagBegin "NewTag"  
  .FormatRuleTag .....  
  .FormatRuleTagColor .....  
  .FormatRuleTagOutline .....  
.FormatRuleTagEnd
```

---

{button ,AL('VENTURA\_rules;;;',0,"Defaultoverview",)} **Related Topics**





# .FormatRuleTagGet (VENTURA)

**.FormatRuleTagGet** .SpaceBefore1=*long*, .SpaceBefore2=*long*, .SpaceBefore3=*long*, .SpaceAfter3=*long*, .Overprint=*long*

This function returns the spacing and overprint attributes of a specified rule tag. To specify the rule tag, enclose the **.FormatRuleTagGet** function with the **.FormatRuleTagBegin** and **.FormatRuleTagEnd** commands.

Syntax	Description
.SpaceBefore1	Specifies a numeric variable that is assigned the space above a rule's first line in tenths of a micron. A rule can be made up of three lines.
.SpaceBefore2	Specifies a numeric variable that is assigned the space above a rule's second line in tenths of a micron. A rule can be made up of three lines.
.SpaceBefore3	Specifies a numeric variable that is assigned the space above a rule's third line in tenths of a micron. A rule can be made up of three lines.
.SpaceAfter3	Specifies a numeric variable that is assigned the space below a rule's third line in tenths of a micron. A rule can be made up of three lines.
.Overprint	Specifies a variable that is assigned a value corresponding to whether a rule has its overprint option enabled: TRUE (-1) if the option is enabled; otherwise, FALSE (0).

### Note

-  This function was introduced in Corel VENTURA 8.
-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example



```
DIM OP AS BOOLEAN
.FormatRuleTagBegin "Sample"
    .FormatRuleTagGet SB1&, SB2&, SB3&, SA&, OP
.FormatRuleTagEnd
```

In above example the **.FormatRuleTagGet** function returns the spacing and overprint attributes for the Sample rule tag's variables **SB1**, **SB2**, **SB3**, **SA**, and **OP**.

## .FormatRuleTagOutline (VENTURA)

**.FormatRuleTagOutline** *.LineIndex=long*, *.Width=long*, *.Type=long*, *.EndCaps=long*, *.JoinType=long*, *.Aspect=long*, *.Angle=long*, *.DotDash=long*, *.RightArrow=long*, *.LeftArrow=long*, *.BehindFill=Boolean*

This command changes the outline attributes of a rule tag. To change a rule tag's attributes, this command must be enclosed by the **.FormatRuleTagBegin** and **.FormatRuleTagEnd** commands.

Syntax	Description
<b>.LineIndex</b>	Specifies which line in the rule to set attributes for. A rule can be made of three lines: 1 First line 2 Second line 3 Third line
<b>.Width</b>	Specifies the width of the specified line in tenths of a micron.
<b>.Type</b>	Specifies the style of the specified line: 0 None 1 Solid 2 Dotted or Dashed  If this parameter is set to Dotted or Dashed, the <b>.DotDash</b> parameter determines the style.
<b>.EndCaps</b>	Specifies the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the width of the nib. A square nib has a value of 100; smaller values make the nib narrower, creating a calligraphy effect. Valid values range from 1 to 100, inclusive.
<b>.Angle</b>	Specifies the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree). This option creates a calligraphic effect.
<b>.DotDash</b>	Specifies the style of the specified line if the <b>.Type</b> parameter is set to Dotted or Dashed. The styles correspond to the Style list box in Corel VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the style of the right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the style of the left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies whether the specified line is placed behind, or in front of, the object's fill. When placed behind, only half the specified line's thickness is visible. This option is particularly useful for outlined text. Set to TRUE (-1) to enable the behind fill option; otherwise, set to FALSE (0), which is the default if omitted.

### Note



This command was introduced in Corel VENTURA 8.



This command corresponds to the Rule Tag Properties dialog box.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatRuleTagBegin "Sample"
.FormatRuleTagOutline 2, 0.05*M_INCH, 2, 0, 0, 90, 25, 7
.FormatRuleTagEnd
```

The above example sets the Sample rule tag's second line to 0.05 inches with a dashed line.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

## .FormatRuleTagOutlineGet (VENTURA)

**.FormatRuleTagOutlineGet** *.LineIndex=long*, *.Width=long*, *.Type=long*, *.EndCaps=long*, *.JoinType=long*, *.Aspect=long*, *.Angle=long*, *.DotDash=long*, *.RightArrow=long*, *.LeftArrow=long*, *.BehindFill=Boolean*

This function returns the outline attributes of a specified rule tag. To specify the rule tag, enclose the **.FormatRuleTagOutlineGet** function with the **.FormatRuleTagBegin** and **.FormatRuleTagEnd** commands.

Syntax	Description
<b>.LineIndex</b>	Specifies which line in the rule to return attributes for. A rule can be made of three lines: 1 First line 2 Second line 3 Third line
<b>.Width</b>	Specifies the numeric variable that is assigned the width of the specified line in tenths of a micron.
<b>.Type</b>	Specifies the numeric variable that is assigned the style of the specified line: 0 None 1 Solid 2 Dotted or Dashed If this parameter assigns Dotted or Dashed, the <b>.DotDash</b> parameter assigns the style.
<b>.EndCaps</b>	Specifies the numeric variable that is assigned the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the numeric variable that is assigned the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the numeric variable that is assigned the width of the nib.
<b>.Angle</b>	Specifies the numeric variable that is assigned the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree).
<b>.DotDash</b>	Specifies the numeric variable that is assigned the style of the specified line if the <b>.Type</b> parameter is set to Dotted or Dashed. The styles correspond to the Style list box in Corel VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the numeric variable that is assigned the style of the right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right) <b>I</b> the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the numeric variable that is assigned the style of the left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right) <b>I</b> the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies the numeric variable that is assigned a value indicating whether the Behind Fill option is enabled: TRUE (-1) if enabled; otherwise, FALSE (0).

### Note

- I** This command was introduced in Corel VENTURA 8.
- I** The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example

```
.FormatRuleTagBegin "Sample"  
  .FormatRuleTagOutlineGet 2, Owidth&, Otype&, , , , Odotdash&  
.FormatRuleTagEnd
```

In above example the **.FormatRuleTagOutlineGet** function returns the outline attributes for the Sample rule tag's second line to the variables **Owidth**, **Otype**, and **Odotdash**.

## .FormatRulingLine (VENTURA)

**.FormatRulingLine** *.RuleIndex=long, .LineIndex=long, .Thickness=long, .SpaceBefore=long, .SpaceAfter=long, .WidthStyle=long, .CustomWidth=long, .CustomIndent=long*

This command has been replaced in Corel VENTURA 8 by [FormatRule](#).

This command modifies a specified ruling line element of the active paragraph or frame. A ruling line element can be made up of three ruling lines. You can also use the [.FormatRulingLineColor](#) and [.FormatRulingLineOutline](#) commands to modify ruling lines.

Syntax	Description
<b>.RuleIndex</b>	Specifies the element to modify: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right 17 Shadow  This element must already exist. Elements are created with the <a href="#">.FormatRulingLineStyle</a> command.
<b>.LineIndex</b>	Specifies the element's line to be modified: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.Thickness</b>	Specifies the thickness of the specified ruling line in tenths of a micron.
<b>.SpaceBefore</b>	Specifies the space before the ruling line in tenths of a micron.
<b>.SpaceAfter</b>	Specifies the space after the third ruling line in tenths of a micron. This parameter should not be used if <b>.LineIndex</b> is set to 1 or 2.
<b>.WidthStyle</b>	Specifies the ruling line width: 0 Paragraph 1 Margin 2 Column 3 Frame 4 Custom
<b>.CustomWidth</b>	If <b>.WidthStyle</b> is set to Custom, specifies the width of the specified ruling line in tenths of a micron.
<b>.CustomIndent</b>	If <b>.WidthStyle</b> is set to Custom, specifies the left indent for the current rule in tenths of a micron.

### Note

**I** This command corresponds to the Ruling Lines dialog box in Corel VENTURA. Click Format, Ruling Lines.

### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatRulingLineStyle 3 'ruling line above and below
.FormatRulingLine 1, 1, .25*M_POINT, 0, 0, 3 'modifies the Above line
.FormatRulingLineColor 1, 1, 2, 0, 40, 60, 0 'modifies the Above line
.FormatRulingLine 2, 1, .25*M_POINT, 0, 0, 3 'modifies the Below line
.FormatRulingLineColor 2, 1, 2, 0, 40, 60, 0 'modifies the Below line
.FormatRulingLineOutline 2, 1, , 2, 0, 0, 90, 25, 7
```

In the above example, the first line specifies to apply ruling lines above and below (1 + 2 = 3) the active paragraph. The second line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The third line modifies the Above line's first line color setting using the CMYK model. The fourth and fifth lines modify the Below line the same way the Above lines were modified. The sixth line modifies the Below line by making it dashed.

The [LENGTHCONVERT](#) function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point.

---


{button ,AL('VENTURA\_rules;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FormatRulingLineColor (VENTURA)


**.FormatRulingLineColor** *.RuleIndex=long, .LineIndex=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long*

This command has been replaced in Corel VENTURA 8 by **FormatRuleColor**.

This command modifies the color of a specified ruling line element of the active paragraph or frame. A ruling line element can be made up of three ruling lines. You can also use the **.FormatRulingLine** and **.FormatRulingLineOutline** commands to modify ruling lines.

Syntax	Description
<b>.RuleIndex</b>	Specifies the element to modify: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right 17 Shadow  This ruling line element must already exist. Elements are created with the <b>.FormatRulingLineStyle</b> command.
<b>.LineIndex</b>	Specifies the element's line to be modified: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.ColorModel</b>	Specifies the color model to use for the specified ruling line: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u>  If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.
<b>.Color1</b>	Specifies the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.
<b>.Color3</b>	Specifies the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.
<b>.Color4</b>	Specifies the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.

### Note

 This command corresponds to the Ruling Lines dialog box in Corel VENTURA. Click Format, Ruling Lines.

### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatRulingLineStyle 3           'ruling line above and below
.FormatRulingLine 1, 1, .25*M_POINT, 0, 0, 3 'modifies the Above line
.FormatRulingLineColor 1, 1, 2, 0, 40, 60, 0 'modifies the Above line
.FormatRulingLine 2, 1, .25*M_POINT, 0, 0, 3 'modifies the Below line
.FormatRulingLineColor 2, 1, 2, 0, 40, 60, 0 'modifies the Below line
.FormatRulingLineOutline 2, 1, , 2, 0, 0, 90, 25, 7
```

In the above example, the first line specifies to apply ruling lines above and below (1 + 2 = 3) the active paragraph. The second line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The third line modifies the Above line's first line color setting using the CMYK model. The fourth and fifth lines modify the Below line the same way the Above lines were modified. The sixth line modifies the Below line by making it dashed.

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point.

---

{button ,AL('VENTURA\_rules;;;',0,"Defaultoverview",)} **Related Topics**

## .FormatRulingLineColorGet (VENTURA)

**.FormatRulingLineColorGet** **.RuleIndex=long**, **.LineIndex=long**, **.ColorModel=long**, **.Color1=long**, **.Color2=long**, **.Color3=long**, **.Color4=long**

This function has been replaced in Corel VENTURA 8 by [FormatRuleColorGet](#).

This function returns the color attributes of a specified ruling line element of the active paragraph or frame. A ruling line element can be made up of three ruling lines. You can also use the [.FormatRulingLineGet](#) and [.FormatRulingLineOutlineGet](#) functions to return ruling line properties.

Syntax	Description
<b>.RuleIndex</b>	Specifies the element to return attributes for: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right 17 Shadow
<b>.LineIndex</b>	Specifies the ruling line to return color attributes for. An element can be made of 3 ruling lines: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.ColorModel</b>	Specifies the numeric variable that is assigned a return value corresponding to the specified ruling line's uniform color model: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
<b>.Color1</b>	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color3</b>	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color4</b>	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatRulingLineColorGet 2, 2, cm&, c1&, c2&, c3&, c4&
```

In the above example the **.FormatRulingLineColorGet** function returns the color attributes for the second ruling line in the Below line element to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

---

{button ,AL(^VENTURA\_rules;;;',0,"Defaultoverview",)} Related Topics



## .FormatRulingLineGet (VENTURA)

**.FormatRulingLineGet** *.RuleIndex=long*, *.LineIndex=long*, *.Thickness=long*, *.SpaceBefore=long*, *.SpaceAfter=long*, *.WidthStyle=long*, *.CustomWidth=long*, *.CustomIndent=long*

This function has been replaced in Corel VENTURA 8 by [FormatRuleGet](#).

This function returns the attributes of a specified ruling line element of a ruling line of the active paragraph or frame. A ruling line element can be made up of three ruling lines. You can also use the [FormatRulingLineColorGet](#) and [FormatRulingLineOutlineGet](#) functions to return ruling line properties.

Syntax	Description
<b>.RuleIndex</b>	Specifies the element to return attributes for: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right 17 Shadow
<b>.LineIndex</b>	Specifies the ruling line to return color attributes for. An element can be made up of three ruling lines: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.Thickness</b>	Specifies the numeric variable that is assigned the thickness of the specified ruling line in tenths of a micron.
<b>.SpaceBefore</b>	Specifies the numeric variable that is assigned the space before the ruling line in tenths of a micron.
<b>.SpaceAfter</b>	Specifies the numeric variable that is assigned the space after the third ruling line in tenths of a micron. This parameter should not be used if <b>.LineIndex</b> is set to 1 or 2.
<b>.WidthStyle</b>	Specifies the numeric variable that is assigned the ruling line width: 0 Paragraph 1 Margin 2 Column 3 Frame 4 Custom
<b>.CustomWidth</b>	If <b>.WidthStyle</b> is set to Custom, specifies the numeric variable that is assigned the width of the specified ruling line in tenths of a micron.
<b>.CustomIndent</b>	If <b>.WidthStyle</b> is set to Custom, specifies the numeric variable that is assigned the left indent for the current rule in tenths of a micron.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatRulingLineGet 2, 2, Thick1&, SpaceBefore&
```

In the above example, the **.FormatRulingLineGet** function returns the attributes for the second ruling line in the Below line element to the variables **Thick1** and **SpaceBefore**.

---



{button ,AL('VENTURA\_rules;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FormatRulingLineOutline (VENTURA)

**.FormatRulingLineOutline** **.RuleIndex=long**, **.LineIndex=long**, **.Width=long**, **.Type=long**, **.EndCaps=long**, **.JoinType=long**, **.Aspect=long**, **.Angle=long**, **.DotDash=long**, **.RightArrow=long**, **.LeftArrow=long**, **.BehindFill=Boolean**

This command has been replaced in Corel VENTURA 8 by **FormatRuleOutline**.

This command modifies the pen settings for a specified ruling line element of the active paragraph or frame. A ruling line element can be made up of three ruling lines. You can also use the **FormatRulingLine** and **FormatRulingLineColor** commands to modify ruling lines.

Syntax	Description
<b>.RuleIndex</b>	Specifies the element to modify: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right 17 Shadow  This element must already exist. Elements are created with the <b><u>FormatRulingLineStyle</u></b> command.
<b>.LineIndex</b>	Specifies the element's line to be modified: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.Width</b>	Specifies the thickness of the specified ruling line in tenths of a micron.
<b>.Type</b>	Specifies the style of the specified ruling line: 0 None 1 Solid 2 Dotted or Dashed  If this parameter is set to Dotted or Dashed, the <b>.DotDash</b> parameter determines the style.
<b>.EndCaps</b>	Specifies the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the width of the nib. A square nib has a value of 100; smaller values make the nib narrower, creating a calligraphy effect. Valid values range from 1 to 100, inclusive.
<b>.Angle</b>	Specifies the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree). This option creates a calligraphic effect.
<b>.DotDash</b>	Specifies the style of the specified ruling line if <b>.Type</b> is set to Dotted or Dashed. The styles correspond to the Style list box in Corel VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the style of the right arrow. Right-arrow styles are listed in the right arrow list box of Corel VENTURA's Outline Pen dialog box. The styles are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the style of the left arrow. Left-arrow styles are listed in the left arrow list box of Corel VENTURA's Outline Pen dialog box. The styles are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies whether the specified ruling line is placed behind, or in front of, the object's fill. When placed behind, only half the specified ruling line's thickness is visible. This option is particularly useful for outlined text. Set to TRUE (-1) to enable the Behind Fill option; otherwise, set to FALSE (0), which is the default if omitted.

### Note



This command corresponds to the Outline Pen dialog box in Corel VENTURA. Click Format, Manage Tag List, Border.

Select a tag and click Edit Tag. Click the pen icon corresponding to the ruling line to edit.

#### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatRulingLineStyle 3           'ruling line above and below
.FormatRulingLine 1, 1, .25*M_POINT, 0, 0, 3 'modifies the Above line
.FormatRulingLineColor 1, 1, 2, 0, 40, 60, 0 'modifies the Above line
.FormatRulingLine 2, 1, .25*M_POINT, 0, 0, 3 'modifies the Below line
.FormatRulingLineColor 2, 1, 2, 0, 40, 60, 0 'modifies the Below line
.FormatRulingLineOutline 2, 1, , 2, 0, 0, 90, 25, 7
```

In the above example, the first line specifies to apply ruling lines above and below ( $1 + 2 = 3$ ) the active paragraph. The second line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The third line modifies the Above line's first line color setting using the CMYK model. The fourth and fifth lines modify the Below line the same way the Above lines were modified. The sixth line modifies the Below line by making it dashed.

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point.

---



{button ,AL('VENTURA\_rules;;;','0,"Defaultoverview",)} Related Topics

## .FormatRulingLineOutlineGet (VENTURA)

**.FormatRulingLineOutlineGet .RuleIndex=long, .LineIndex=long, .Width=long, .Type=long, .EndCaps=long, .JoinType=long, .Aspect=long, .Angle=long, .DotDash=long, .RightArrow=long, .LeftArrow=long, .BehindFill=Boolean**

This function has been replaced in Corel VENTURA 8 by [FormatRuleOutlineGet](#).

This function returns the pen settings for a specified ruling line element of the active paragraph or frame. A ruling line element can be made up of three ruling lines. You can also use the [.FormatRulingLineGet](#) and [.FormatRulingLineColorGet](#) functions to return ruling line properties.

Syntax	Description
<b>.RuleIndex</b>	Specifies the element to return attributes for: 0 None 1 Above 2 Below 4 Box 8 Left 16 Right 17 Shadow
<b>.LineIndex</b>	Specifies the ruling line to return attributes for. An element can be made up of three ruling lines: 1 First ruling line 2 Second ruling line 3 Third ruling line
<b>.Width</b>	Specifies the numeric variable that is assigned the width of the specified ruling line in tenths of a micron.
<b>.Type</b>	Specifies the numeric variable that is assigned the style of the specified ruling line: 0 None 1 Solid 2 Dotted or Dashed If this parameter assigns Dotted or Dashed, the <b>.DotDash</b> parameter assigns the specific style.
<b>.EndCaps</b>	Specifies the numeric variable that is assigned the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the numeric variable that is assigned the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the numeric variable that is assigned the width of the nib.
<b>.Angle</b>	Specifies the numeric variable that is assigned the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree).
<b>.DotDash</b>	Specifies the numeric variable that is assigned the style of the specified ruling line if <b>.Type</b> is set to Dotted or Dashed. The styles correspond to the Style list box in Corel VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the numeric variable that is assigned the style of the right arrow. Right-arrow styles are listed in the right arrow list box of VENTURA's Outline Pen dialog box. The styles are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the numeric variable that is assigned the style of the left arrow. Left-arrow styles are listed in the left arrow list box of VENTURA's Outline Pen Roll-Up. The styles are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies the numeric variable that is assigned a value indicating whether the Behind Fill option is enabled. If the option is enabled, TRUE (-1) is assigned; otherwise, FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

#### Example

```
.FormatRulingLineOutlineGet 2, 2, Owidth&, Otype&, , , , , Odotdash&
```

In the above example the **.FormatRulingLineOutlineGet** function returns the attributes for the second ruling line in the Below line element to the variables **Owidth**, **Otype**, and **Odotdash**.

---

{button ,AL('VENTURA\_rules;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.FormatRulingLineStyle (VENTURA)**

**.FormatRulingLineStyle** .RuleStyle=*long*, .PresetRule=*string*

This command has been replaced in Corel VENTURA 8 by **FormatRuleStyle**.

This command specifies the ruling lines to apply to the active paragraph or frame. This command should be used in conjunction with the **FormatRulingLine**, **FormatRulingLineColor**, and **FormatRulingLineOutline** commands which both set properties for ruling lines.

Syntax	Description														
.RuleStyle	<p>Specifies the ruling line elements to apply:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Above</td></tr><tr><td>2</td><td>Below</td></tr><tr><td>4</td><td>Box</td></tr><tr><td>8</td><td>Left</td></tr><tr><td>16</td><td>Right</td></tr><tr><td>17</td><td>Shadow</td></tr></table> <p>Elements can be combined by adding their corresponding values. For Above &amp; Below, set this parameter to 3.</p>	0	None	1	Above	2	Below	4	Box	8	Left	16	Right	17	Shadow
0	None														
1	Above														
2	Below														
4	Box														
8	Left														
16	Right														
17	Shadow														
.PresetRule	<p>Specifies the name of a ruling line style preset on the user's system. The available names can be found in the Ruling Lines dialog box (Advanced Tab) in the Presets list box. If this parameter is used, the <b>.RuleStyle</b> parameter is ignored.</p>														

### **Note**

**I** This command corresponds to the Ruling Lines dialog box in Corel VENTURA. Click Format, Ruling Lines.

### **Example**

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatRulingLineStyle 3           'ruling line above and below
.FormatRulingLine 1, 1, .25*M_POINT, 0, 0, 3 'modifies the Above line
.FormatRulingLine 1, 1, 2, 0, 40, 60, 0 'modifies the Above line
.FormatRulingLine 2, 1, .25*M_POINT, 0, 0, 3 'modifies the Below line
.FormatRulingLineColor 2, 1, 2, 0, 40, 60, 0 'modifies the Below line
.FormatRulingLineOutline 2, 1, , 2, 0, 0, 90, 25, 7
```

In the above example, the first line specifies to apply ruling lines above and below ( $1 + 2 = 3$ ) the active paragraph. The second line modifies the Above line's first line, setting it to a width of 0.25 points, and setting it to frame wide. The third line modifies the Above line's first line color setting using the CMYK model. The fourth and fifth lines modify the Below line the same way the Above lines were modified. The sixth line modifies the Below line by making it dashed.

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point.

---

{button ,AL('VENTURA\_rules;;;','0','Defaultoverview"),}} **Related Topics**

## .FormatRulingLineStyleGet (VENTURA)

**.FormatRulingLineStyleGet** .RuleStyle=*long*, .PresetRule=*string*

This function has been replaced in Corel VENTURA 8 by [FormatRuleStyleGet](#).

This function returns which ruling lines are applied to the active paragraph or frame.

Syntax	Description														
.RuleStyle	<p>Specifies the numeric variable that is assigned the ruling lines that are applied:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Above</td></tr><tr><td>2</td><td>Below</td></tr><tr><td>4</td><td>Box</td></tr><tr><td>8</td><td>Left</td></tr><tr><td>16</td><td>Right</td></tr><tr><td>17</td><td>Shadow</td></tr></table> <p>Elements can be combined by adding their corresponding values. For example, if Above &amp; Below are both enabled, this parameter returns 3.</p>	0	None	1	Above	2	Below	4	Box	8	Left	16	Right	17	Shadow
0	None														
1	Above														
2	Below														
4	Box														
8	Left														
16	Right														
17	Shadow														
.PresetRule	<p>Specifies the string variable that is assigned the name of a ruling line style preset applied to the active paragraph or frame (if it exists). The preset names can be found in the Presets list box of Corel VENTURA's the Ruling Lines dialog box (Advanced Tab).</p>														

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatRulingLineStyleGet RuleStyle&
```

In the above example the function returns which ruling lines are applied to the active paragraph or frame to the **RuleStyle** variable.

In cases where multiple elements are returned, you can use the AND ([bitwise](#)) operator to determine specific elements. The following example shows how to test if the variable used in the above example includes the Left element:

```
IF 8 AND RuleStyle& THEN LeftTest$ = "Yes" ELSE LeftTest$ = "No"
```

8 is the value of Left element. The variable **LeftTest** is assigned a string based on bitwise comparison. In this example, **LeftTest** is assigned **Yes** if **RuleStyle** has the Left element.

---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} [Related Topics](#)

## .RuleTagAddNew (VENTURA)

**.RuleTagAddNew .TagName=*string*, .CopyFrom=*string***

This command adds a new rule tag to the current publication. The new tag is based on an existing rule tag.

Syntax	Description
<b>.TagName</b>	Specifies the name of the new rule tag. The tag name must not already exist in the style sheet.
<b>.CopyFrom</b>	Specifies the rule tag that the new rule tag is based on.

### Note



This command corresponds to the Tag Window. Click Tools, Tag Window.



This command was introduced in Core! VENTURA 8.

### Example

```
.RuleTagAddNew "2P_Special", "2P_SOLID"
```

The above example adds a rule tag named 2P\_Special to the current publication. The new tag is based on the 2P\_SOLID tag.

---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} Related Topics



## .RuleTagCopy (VENTURA)

**.RuleTagCopy** .TagName=*string*

This command copies a rule tag from the current style sheet to the Clipboard. This command can be used to copy tags into VENTURA Libraries.

Syntax	Description
<b>.TagName</b>	Specifies the name of the rule tag to copy.

### Note



This command corresponds to the Tag Window. Click Tools, Tag Window.



This command was introduced in Corel VENTURA 8.

### Example

```
.RuleTagCopy "TradeMark"  
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste
```

The above example copies the rule tag TradeMark, creates a VENTURA library named MYLIB.VLB, and pastes the TradeMark tag into the new VENTURA library.

---

{button ,AL("VENTURA\_rules;;;','0,"Defaultoverview"),} Related Topics

## .RuleTagCount (VENTURA)

**ReturnValue**& = .RuleTagCount ( )

This function returns the number of rule tags in the current style sheet.

Syntax	Description
<b>ReturnValue</b> &	Specifies the numeric variable that is assigned the number of rule tags in the current style sheet.

### Note

**I** This function was introduced in Corel VENTURA 8.

### Example

```
Rcounts = .RuleTagCount ( )
```


The number of rule tags is assigned the **Rcounts** variable.


---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview",)} [Related Topics](#)

## .RuleTagDelete (VENTURA)




**.RuleTagDelete .TagName=string, .ReformatTagName=string**

This command deletes a rule tag from the current style sheet. Any item in the publication  and in other publications that are linked to the same style sheet through the VENTURA Library

 formatted with the deleted tag will be reformatted using another tag of your choice.

Syntax	Description
<b>.TagName</b>	Specifies the name of the rule tag to delete.
<b>.ReformatTagName</b>	Specifies the rule tag to apply to borders tagged with <b>.TagName</b> .

### Note

-  This command corresponds to the Tag Window. Click Tools, Tag Window.
-  This command was introduced in Corel VENTURA 8.
-  You cannot delete the following rule tags: Double, Hidden, Single, and Thick.

### Example

```
.RuleTagDelete "Montreal", "Boston"
```

The above example deletes the rule tag Montreal and applies the Boston tag to rules using the Montreal tag.

---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} [Related Topics](#)

## .RuleTagGetAt (VENTURA)

**ReturnString\$ = .RuleTagGetAt .TagIndex=*long***

This function returns the rule tag name associated with a rule tag index number in the current style sheet.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of a rule tag.
<b>.TagIndex</b>	Specifies the index number of a rule tag. Index numbers are not associated with tag order in the Manage Tag List dialog box (alphabetical order). Instead, index numbers are associated by the order in which the rule tags are created. The first tag created is 1, the second tag created is 2, and so on. The last created tag is equal to the return value from the <b>.RuleTagCount</b> function. If a tag is deleted, the tag index is recompiled.

### Note



This function was introduced in Corel VENTURA 8.

### Example

```
LastTag& = .RuleTagCount ( )      'number of tags  
LastTagName$ = .RuleTagGetAt (LastTag&)
```

In the above example, the first statement counts the number of rule tags. The second statement is assigned the name of last created rule tag still available in the current style sheet.

---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview"),} [Related Topics](#)

## .RuleTagRename (VENTURA)

**.RuleTagRename .OldTagName=*string*, .NewTagName=*string***

This command renames a specified rule tag. Renaming tags may affect the formatting of other publications that are linked to the same style sheet through the Library.

Syntax	Description
<b>.OldTagName</b>	Specifies the rule tag to be renamed.
<b>.NewTagName</b>	Specifies the new rule tag name.

### Note



This command corresponds to the Tag Window. Click Tools, Tag Window.



This command was introduced in Core! VENTURA 8.



You cannot rename the following rule tags: Double, Hidden, Single, and Thick.

### Example

```
.RuleTagRename "Name 1", "Name 2"
```

The above example renames the rule tag Name 1 to Name 2.

---

{button ,AL('VENTURA\_rules;;;','0',"Defaultoverview",)} Related Topics


## .ChapterAdd (VENTURA)

**.ChapterAdd** *.ChapterName=string*

This command adds an existing VENTURA version 3.x, 4.x, or 5 chapter (.CHP) to the active publication.

Syntax	Description
<b>.ChapterName</b>	Specifies the name and path of the file to add. If the path is not specified, the active folder is used.

### Note

 This command corresponds to right-clicking Import Chapter while the VENTURA Navigator is in Publication Manager mode in Corel VENTURA.

### Example

```
.ChapterAdd "C:\Corel50\Ventura\Overview.CHP"
```

The above example adds the OVERVIEW.CHP chapter to the active publication.

---

**{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),}** [Related Topics](#)


## **.ChapterAddNew (VENTURA)**

**.ChapterAddNew .ChapterName=*string***

This command adds a new chapter to the active publication.

Syntax	Description
<b>.ChapterName</b>	Specifies the name of the new chapter.

### **Note**

 This command corresponds to right-clicking New Chapter while the Navigator is in Publication Manager mode in Corel VENTURA.

### **Example**

**.ChapterAddNew "Quarter"**

The above example adds a new chapter titled Quarter to the active publication.

---

**{Button,ALink(VENTURA\_Chapter;;;,0,"Defaultoverview",)} Related Topics**

## **.ChapterCopy (VENTURA)**

### **.ChapterCopy**

This command copies the active chapter to the Windows clipboard.

#### **Note**



This command cannot be recorded.



You can use the **ChapterPaste** command to paste a copied chapter into another publication.



This command corresponds to right-clicking Copy while the Navigator is in Publication Manager mode in Corel VENTURA.

#### **Example**

`.ChapterCopy`

---

`{Button,ALink(VENTURA_Chapter;;;;;0,"Defaultoverview"),}` **Related Topics**



## .ChapterCount (VENTURA)

**ReturnValue& = .ChapterCount ( )**

This function returns the number of chapters in the active publication.

Syntax	Description
<b>ReturnValue&amp;</b>	The numeric variable that is assigned the number of chapters in the active publication.

### Note



This command cannot be recorded.

### Example

```
CCount& = .ChapterCount ( )
```

The above example assigns the number of chapters in the active publication to the **CCount** variable.

---

**{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),}** [Related Topics](#)


## .ChapterDelete (VENTURA)

**.ChapterDelete** *.ChapterName=string*

This command removes an existing chapter from the active publication.

Syntax	Description
<b>.ChapterName</b>	Specifies the name of the chapter to remove. This name is case sensitive.

### Note

 This command corresponds to right-clicking Delete while the Navigator is in Publication Manager mode in Corel VENTURA.

### Example

```
.ChapterDelete "Old Version"
```

The above example removes the Old Version chapter from the active publication.

---

`{Button,ALink(VENTURA_Chapter;;;,0,"Defaultoverview"),}` [Related Topics](#)

## **.ChapterGetAt (VENTURA)**

**ReturnString\$ = .ChapterGetAt .ChapterIndex=*long***

This function returns the chapter name associated with a chapter index number in the active publication.

<b>Syntax</b>	<b>Description</b>
<b>ReturnString\$</b>	Specifies the string variable that is assigned the chapter name.
<b>.ChapterIndex</b>	Specifies the chapter index number. These numbers correspond to the order of the chapters in the Navigator's Publication Manager. First listed chapter is 1, the second is 2, and so on. The last listed chapter number is equal to the return value from the <a href="#"><b>.ChapterCount</b></a> function.

### **Note**



This command cannot be recorded.

### **Example**

```
CCount& = .ChapterCount ( ) 'number of chapters  
LastChapterName$ = .ChapterGetAt (CCount&)
```

In the above example, the first statement counts the number of chapters. The second statement is assigned the name of last chapter in the active publication.

---

**{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),}** [\*\*Related Topics\*\*](#)

## .ChapterPageCount (VENTURA)

**ReturnValue** = .ChapterPageCount ( )

This function returns the number of pages in the current chapter.

Syntax	Description
<b>ReturnValue</b>	The numeric variable that is assigned the number of pages in the current chapter.

### Note



This command cannot be recorded.

### Example

```
CPCount & = .ChapterPageCount ( )
```

The above example assigns the number of pages in the current chapter to the **CPCount** variable.

---

{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),} Related Topics

## **.ChapterPaste (VENTURA)**

### **.ChapterPaste**

This command pastes a chapter from the Windows clipboard to another publication or a VENTURA Library.

#### **Note**



This command cannot be recorded, and it was introduced in Corel VENTURA 8.



You can use the **ChapterCopy** command to copy the active publication.



This command corresponds to right-clicking Paste while the VENTURA Navigator is in Publication Manager mode in Corel VENTURA.

#### **Example**

`.ChapterPaste`

---

`{Button,ALink(VENTURA_Chapter;;;;,0,"Defaultoverview"),}` Related Topics


## .ChapterRename (VENTURA)

**.ChapterRename .OldChpName=*string*, .NewChpName=*string***

This command renames a specified chapter in the active publication.

Syntax	Description
<b>.OldChpName</b>	Specifies the chapter to be renamed. This name is case sensitive.
<b>.NewChpName</b>	Specifies the new name for the specified chapter.

### Note

 This command corresponds to right-clicking Rename while the Navigator is in Publication Manager mode in Corel VENTURA.

### Example

```
.ChapterRename "Chp Name 1" "Chp Name 2"
```

The above example renames the chapter Chp Name 1 to Chp Name 2.

---

**{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),}** [Related Topics](#)

## .CurrentChapter (VENTURA)

ReturnValue& =.CurrentChapter ( )

This function returns the name of the active chapter.

Syntax	Description
ReturnString\$	Specifies the string variable that is assigned the name of the active chapter.

### Note



This command cannot be recorded.



The chapter name this function returns corresponds to the name displayed in the VENTURA Navigator in Publication Manager mode.

### Example

```
Cname$ = .CurrentChapter ( )
```

In the above example, the **CName** variable is assigned the name of the active chapter.

---

{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),} Related Topics


# .FormatAutoNumberDefine (VENTURA)

**.FormatAutoNumberDefine** *.LevelNumber=long*, *.PrefixText=string*, *.NumberStyle=long*, *.ParagraphTag=string*, *.SuffixText=string*, *.StartNumber=long*, *.IncludePrevLevel=Boolean*

This command sets the properties for the automatic paragraph numbers in the active chapter. Use the **.FormatChapterAutoNumber** command to set the automatic numbering properties for paragraphs in the active chapter.

Syntax	Description
<b>.LevelNumber</b>	Specifies the level number to set properties for. This level must exist in the chapter as defined in the <b>.FormatChapterAutoNumber</b> command.
<b>.PrefixText</b>	Specifies the text to insert before the specified level number.
<b>.NumberStyle</b>	Specifies the counter numbering style: 0 1,2,3...(default) 1 A,B,C... 2 a,b,c... 3 I, II, III... 4 i, ii, iii... 5 One, Two, Three... 6 ONE, TWO, THREE... 7 one, two, three...
<b>.ParagraphTag</b>	Specifies the paragraphs to number based on a paragraph tag.
<b>.SuffixText</b>	Specifies the text to insert after the specified level number.
<b>.StartNumber</b>	Specifies the start number for the specified level.
<b>.IncludePrevLevel</b>	Specifies whether to continue numbering sequentially from the previous chapter. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0) which is the default if omitted.

## Note

 This command corresponds to the Auto-numbering tab on the Chapter Properties dialog box in Corel VENTURA. Click Publication, Chapter Properties, AutoNumbering.

## Example

```
.FormatChapterAutoNumber TRUE, 2
.FormatAutoNumberDefine 1, "", 0, "Body Text", "", 10, FALSE
.FormatAutoNumberDefine 2, "", 4, "Body Text2", "", 20, FALSE
```

In the above example, the first line enables automatic numbering for paragraphs in the active chapter and to use two levels of automatic numbering. The second line sets the properties for the first level, while the third line sets the properties for the second level.



# .FormatAutoNumberGetAt (VENTURA)

**.FormatAutoNumberGetAt** *.LevelNumber=long*, *.PrefixText=string*, *.NumberStyle=long*, *.ParagraphTag=string*, *.SuffixText=string*, *.StartNumber=long*, *.IncludePrevLevel=Boolean*

This command gets the values for the specific auto-numbering level.

Syntax	Description
<b>.LevelNumber</b>	Specifies the level number to return properties for. This level must exist in the chapter as defined in the <b>FormatChapterAutoNumber</b> command.
<b>.PrefixText</b>	Specifies a string variable which is assigned the text to insert before the specified level number.
<b>.NumberStyle</b>	Specifies a numeric variable which is assigned the counter numbering style: 0 1,2,3...(default) 1 A,B,C... 2 a,b,c... 3 I, II, III... 4 i, ii, iii... 5 One, Two, Three... 6 ONE, TWO, THREE... 7 one, two, three...
<b>.ParagraphTag</b>	Specifies a string variable which is assigned the paragraphs to number based on a paragraph tag.
<b>.SuffixText</b>	Specifies a string variable which is assigned the text to insert after the specified level number.
<b>.StartNumber</b>	Specifies a numeric variable which is assigned the start number for the specified level.
<b>.IncludePrevLevel</b>	Specifies a numeric variable which is assigned the continue numbering sequentially from the previous chapter setting: TRUE (-1) if enabled; otherwise FALSE (0).

## Note

- I** The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
- I** This command cannot be recorded.

## Example

`.FormatAutoNumberGetAt 1, PText$, NStyle&`

This example returns the prefix text and numbering style attributes for the first auto-numbering level to the variables **PText** and **NStyle**.

---

`{Button,ALink(VENTURA_Chapter;;;,0,"Defaultoverview"),}` **Related Topics**


## .FormatChapterAutoNumber (VENTURA)


**.FormatChapterAutoNumber** *.Enable=Boolean, .NumberOfLevels=long, .AcrossChapters=Boolean*

This command sets the automatic numbering properties of paragraphs in the active chapter.

Syntax	Description
<b>.Enable</b>	Specifies whether to enable or disable auto-numbering. Set to TRUE (-1) to enable. Set to FALSE (0) to disable.
<b>.NumberOfLevels</b>	Specifies the number of levels to auto-number. The maximum number of levels is 10.
<b>.AcrossChapters</b>	Specifies whether to enable or disable auto-numbering across chapters. Set to TRUE (-1) to enable. Set to FALSE (0) to disable.

### Note

 This command corresponds to the AutoNumbering tab on the Chapter Properties dialog box in Corel VENTURA. Click Publication, Chapter Properties, AutoNumbering.

 Use the **.FormatAutoNumberDefine** command to set the properties for the automatic numbers.

### Example

```
.FormatChapterAutoNumber TRUE, 2
.FormatAutoNumberDefine 1, "*", 0, "Body Text", "*", 10, FALSE
.FormatAutoNumberDefine 2, "*", 4, "Body Text2", "*", 20, FALSE
```

In the above example, the first line enables automatic numbering for paragraphs in the active chapter and to use two levels of automatic numbering. The second line sets the properties for the first level, while the third line sets the properties for the second level.

---

**{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),}** [Related Topics](#)

## **.FormatChapterAutoNumberGet (VENTURA)**

**.FormatChapterAutoNumberGet** .Enable=*Boolean*, .NumberOfLevels=*long*, .AcrossChapters=*Boolean*

This function returns the paragraph automatic numbering properties in the active chapter.

Syntax	Description
.Enable	Specifies a numeric variable which is assigned the auto-numbering setting: TRUE (-1) if enabled; otherwise FALSE (0).
.NumberOfLevels	Specifies a numeric variable which is assigned the number of levels to auto-number.
.AcrossChapters	Specifies a numeric variable which is assigned the auto-numbering across chapters setting: TRUE (-1) if enabled; otherwise FALSE (0).

### **Note**



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### **Example**

```
DIM En AS BOOLEAN
DIM AcrossChapters AS BOOLEAN
.FormatChapterAutoNumberGet En, NLevels&, AChapters
```

This example returns the auto-numbering attributes of the current chapter to the variables **En**, **NLevels**, and **AChapters**.

---

**{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),} Related Topics**

## .FormatChapterCounter (VENTURA)

**.FormatChapterCounter** **.CounterType=long**, **.ContinueFromPrev=Boolean**, **.StartNumber=long**, **.NumberStyle=long**

This command sets the format and starting number of counters for pages, figures, tables, and chapters.

Syntax	Description
<b>.CounterType</b>	Specifies the type of counter to format: 0 Initial Page 1 Initial Table 2 Initial Figure 3 Chapter
<b>.ContinueFromPrev</b>	Specifies whether to continue numbering from the previous counter specified in <b>.CounterType</b> . Set to TRUE (-1) to enable this option; otherwise set to FALSE (0). If omitted, set to FALSE.
<b>.StartNumber</b>	If <b>.ContinueFromPrev</b> is set to FALSE, specifies the starting number for the specified counter.
<b>.NumberStyle</b>	Specifies the counter numbering style: 0 1,2,3...(default) 1 A,B,C... 2 a,b,c... 3 I, II, III... 4 i, ii, iii... 5 One, Two, Three... 6 ONE, TWO, THREE... 7 one, two, three...

### Note

**I** This command corresponds to the Counters tab on the Chapter Properties dialog box in Corel VENTURA. Click Publication, Chapter Properties, Counters.

### Example

```
.FormatChapterCounter 3, 0, 12, 6
```

The above example sets the number for the active chapter to 12 using the ONE, TWO, THREE... style.

---

**{Button,ALink(VENTURA\_Chapter;;;,0,"Defaultoverview"),}** **Related Topics**



# .FormatChapterCounterGet (VENTURA)

**.FormatChapterCounterGet .CounterType=long, .ContinueFromPrev=Boolean, .StartNumber=long, .NumberStyle=long**

This function returns the format and starting number of counters for pages, figures, tables, and chapters.

Syntax	Description
.CounterType	Specifies the type of counter to return values for: 0 Initial Page 1 Initial Table 2 Initial Figure 3 Chapter
.ContinueFromPrev	Specifies a numeric variable which is assigned the continue numbering from the previous counter setting: TRUE (-1) if enabled; otherwise FALSE (0).
.StartNumber	If <b>.ContinueFromPrev</b> is set to FALSE (0), specifies a numeric variable which is assigned the starting number for the specified counter.
.NumberStyle	Specifies a numeric variable which is assigned the counter numbering style: 0 1,2,3...(default) 1 A,B,C... 2 a,b,c... 3 I, II, III... 4 i, ii, iii... 5 One, Two, Three... 6 ONE, TWO, THREE... 7 one, two, three...

**Note**

- The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
- This command cannot be recorded.

**Example**

```
DIM ContPrev AS BOOLEAN
.FormatChapterCounterGet 3, ContPrev, SNumber&, NStyle&
```

This example gets the setting for the chapter counter.

---

**{Button,ALink(VENTURA\_Chapter;;;;0,"Defaultoverview"),}** **Related Topics**


**.FormatChapterEndnote (VENTURA)**

**.FormatChapterEndnote** .Enable=*Boolean*, .StartNumber=*long*, .NumberStyle=*long*, .Position=*Boolean*, .RestartOn=*Boolean*, .CustomMark=*string*, .Format=*string*

This command enables and disables endnotes in the active chapter. It also adjusts note placement, sets numbering options, and adds (or removes) the line separating the main text from the notes.

Syntax	Description
.Enable	Specifies whether to enable endnotes in the active chapter. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If this option is not set to TRUE, all other parameters are ignored.
.StartNumber	Specifies the number for the first automatically numbered note.
.NumberStyle	Specifies the endnote numbering style: 0 1,2,3...(default) 1 A,B,C... 2 a,b,c... 3 I, II, III... 4 i, ii, iii... 5 One, Two, Three... 6 ONE, TWO, THREE... 7 one, two, three...
.Position	Set this parameter to 0.
.RestartOn	Set this parameter to 0.
.CustomMark	Specifies the character to use to number notes. If this parameter is used the .NumberStyle parameter is ignored. Custom numbering schemes can consist of eight character strings delimited by a space. For example, @ ## \$ %% ^^ & * !. If the number of notes exceeds the number of character strings in the numbering scheme, VENTURA starts renumbering with the first character string.
.Format	Specifies the format of the endnote mark. The following are typical examples of endnote marks (the # indicates a number): # (default) [#] {#} (#) -#- #.

**Note**

 This command corresponds to the Endnotes tab on the Chapter Properties dialog box in Corel VENTURA. Click Publication, Chapter Properties, Endnotes.

**Example**

`.FormatChapterEndnote TRUE, 5, 1, .Format= "[#]"`

The above example enables endnotes in the active chapter. The starting number is set to 5, the A, B, C, ... number format is used, and the endnote marker is enclosed with square brackets.



# .FormatChapterEndnoteGet (VENTURA)

**.FormatChapterEndnoteGet** .Enable=*Boolean*, .StartNumber=*long*, .NumberStyle=*long*, .Position=*long*, .RestartOn=*long*, .CustomMark=*string*, .Format=*string*

This function returns chapter endnote settings.

Syntax	Description
.Enable	Specifies a numeric variable which is assigned a value corresponding to the status of endnotes in the active chapter: TRUE (-1) if enabled; otherwise, FALSE (0).
.StartNumber	Specifies a numeric variable which is assigned the number for the first automatically numbered note.
.NumberStyle	Specifies a numeric variable which is assigned the endnote numbering style: 0 1,2,3... 1 A,B,C... 2 a,b,c... 3 I, II, III... 4 i, ii, iii... 5 One, Two, Three... 6 ONE, TWO, THREE... 7 one, two, three...
.Position	This parameter is not valid in Corel VENTURA 8. It is used as a placeholder.
.RestartOn	This parameter is not valid in Corel VENTURA 8. It is used as a placeholder.
.CustomMark	Specifies a string variable which is assigned a value corresponding to the character numbering scheme used in the chapter (if enabled).
.Format	Specifies a string variable which is assigned the format of the endnote mark. The following are typical examples of endnote marks: # (default) [#] {#} (#) -#- #.

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

## Example

```
DIM En AS BOOLEAN
.FormatChapterEndnoteGet En, StartNumber&, NumberStyle&
```

This example assigns the endnote attributes for the current chapter to the variables **En**, **StartNumber**, and **NumberStyle**.


## .FormatChapterFootnote (VENTURA)


**.FormatChapterFootnote** .Enable=*Boolean*, .StartNumber=*long*, .Position=*long*, .NumberStyle=*long*, .CustomMark=*string*, .Format=*long*, .RestartOn=*long*, .FlowInColumn=*Boolean*, .KeepWithRef=*Boolean*, .UseMaxHeight=*Boolean*, .MaxFrameHeight=*long*

This command enables or disables footnotes in the active chapter. It also adjusts note placement, sets numbering options, and adds (or removes) the line separating the main text from the notes.

Syntax	Description
.Enable	Specifies whether to enable footnotes in the chapter. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If this option is not set to TRUE, all other parameters are ignored.
.StartNumber	Specifies the number for the first automatically numbered note.
.Position	Set this parameter to 0.
.NumberStyle	Specifies the footnote numbering style: 0 1,2,3...(default) 1 A,B,C... 2 a,b,c... 3 I, II, III... 4 i, ii, iii... 5 One, Two, Three... 6 ONE, TWO, THREE... 7 one, two, three...
.CustomMark	Specifies the character to use to number notes. If this parameter is used, the <b>.NumberStyle</b> parameter is ignored. Custom numbering schemes can consist of eight character strings delimited by a space; for example, @ ## \$ %% ^^ & * !!. If the number of notes exceeds the number of character strings in the numbering scheme, Corel VENTURA starts renumbering with the first character string.
.Format	Specifies the format of the footnote mark (the # indicates a number): 0 # (default) 1 [#] 2 {#} 3 (#) 4 -#- 5 #.
.RestartOn	Specifies whether to restart numbering on each page or continue in sequence throughout the chapter: 0 On page 1 On chapter
.FlowInColumn	Specifies whether to keep footnotes in a multi-column publication within the same column as the footnote reference mark. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.KeepWithRef	Specifies whether to always keep the footnote and its reference on the same page even if causes widows and orphans or large gaps to appear between the footnote and the main text. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.UseMaxHeight	Specifies whether to use the Maximum Height option. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). The maximum height is set in the <b>.MaxFrameHeight</b> parameter.
.MaxFrameHeight	Specifies the amount of space you want to set aside for footnote text, in tenths of a micron. Text exceeding the limit continues on the next page. The <b>.UseMaxHeight</b> parameter must be set to TRUE to use this option.

### Notes

 This command corresponds to the Footnotes tab in the Chapter Properties dialog box in Corel VENTURA. Click Publication, Chapter Properties, Footnotes.

 Use the **.FormatFootnoteSeparator** command to set chapter footnote separator properties.

### Example

```
M_INCH = LENGTHCONVERT (1, 7, 1)
M_POINT = LENGTHCONVERT (1, 3, 1)
.FormatChapterFootnote TRUE, 4,,, 1,,, TRUE, 1.25*M_INCH
.FormatFootnoteSeparator TRUE, 1.5*M_POINT, 6.5*M_INCH, 0.2*M_POINT
```



In the above example, the third line enables footnotes in the chapter, setting the maximum height of the footnote text to 1.25 inches. The fourth line sets the properties of the footnote separator line to use in the chapter .

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point and another variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

**{Button,ALink(VENTURA\_Chapter;;;,0,"Defaultoverview",)} Related Topics**

## .FormatChapterFootnoteGet (VENTURA)

**.FormatChapterFootnoteGet** .Enable=*Boolean*, .StartNumber=*long*, .NumberStyle=*long*, .Position=*long*, .RestartOn=*long*, .FlowInColumn=*Boolean*, .KeepWithRef=*Boolean*, .UseMaxHeight=*Boolean*, .MaxFrameHeight=*long*, .CustomMark=*string*, .Format=*string*

This returns the chapter footnote settings.

Syntax	Description
.Enable	Specifies a numeric variable which is assigned a value corresponding to the status of footnotes in the active chapter: TRUE (-1) if enabled; otherwise, FALSE (0).
.StartNumber	Specifies a numeric variable which is assigned the number for the first automatically numbered note.
.NumberStyle	Specifies a numeric variable which is assigned the footnote numbering style: 0 1,2,3...(default) 1 A,B,C... 2 a,b,c... 3 I, II, III... 4 i, ii, iii... 5 One, Two, Three... 6 ONE, TWO, THREE... 7 one, two, three...
.Position	This parameter is not valid in Corel VENTURA 8. It is used as a placeholder.
.RestartOn	Specifies a numeric variable which is assigned a value representing whether to restart numbering on each page or continue in sequence throughout the chapter: 0 On page 1 On chapter
.FlowInColumn	Specifies a numeric variable which is assigned a value corresponding to the status of footnotes in multi-column publications: TRUE (-1) is assigned if footnotes are in the same column; otherwise, FALSE (0).
.KeepWithRef	Specifies a numeric variable which is assigned TRUE (-1) if the footnote and its reference are kept on the same page even if causes widows and orphans or large gaps to appear between the footnote and the main text.
.UseMaxHeight	Specifies a numeric variable which is assigned TRUE (-1) if the Maximum Height option is enabled. If not enabled, FALSE (0) is assigned.
.MaxFrameHeight	Specifies a numeric variable which is assigned the amount of space to set aside for footnote text, in tenths of a micron. Text exceeding the limit continues on the next page.
.CustomMark	Specifies a string variable which is assigned a value corresponding to the character numbering scheme used in the chapter (if enabled).
.Format	Specifies a numeric variable which is assigned the format of the endnote mark. The following are typical examples of endnote marks (the # indicates a number): 0 # (default) 1 [#] 2 {#} 3 (#) 4 -#- 5 #.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM En AS BOOLEAN
.FormatChapterFootnoteGet En, SNumber&, Nstyle&
```

This example assigns the footnote attributes for the current chapter to the variables **En**, **SNumber**, and **NStyle**.

---

{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),} Related Topics

## .FormatChapterGeneral (VENTURA)

**.FormatChapterGeneral** .LayoutType=*long*, .SingleSided=*Boolean*, .StartOnLeft=*Boolean* .Condition=*string* .VJParagraphTag=*string* .VJCustomAmount=*long* .HangingPunctuation=*string* .StartOnEither=*Boolean*

This command sets the chapter layout properties for the active chapter.

Syntax	Description
.LayoutType	Set this parameter to 0. This parameter is no longer used in Corel VENTURA 8.
.SingleSided	Specifies whether publication has a single or double-sided layout. Set to TRUE (-1) for single-sided. Set to FALSE (0) for double-sided. The default is set to FALSE.
.StartOnLeft	Specifies whether to start the publication on a right or left page. Set to TRUE (-1) start on the left side. Set to FALSE (0) to start on the right. If omitted, set to FALSE.
.Condition	String that specifies the condition of the chapter.
.VJParagraphTag	Specifies the paragraph tag that contains the interline spacing for the vertically justified publication.
.VJCustomAmount	Specifies the custom interline spacing value.
.HangingPunctuation	String that contains the list of hanging punctuation. One or more characters followed by the overhang percentage in square brackets. For example, ";;.[50]!?,[100]^[75]".
.StartOnEither	Specifies the starting page side option. Set to TRUE (-1) to start on either side depending on where the previous chapter ends.

### Note

**I** This command corresponds to the General tab on the Chapter Properties dialog box in Corel VENTURA. Click Publication, Chapter Properties, General.

### Example

```
.FormatChapterGeneral 0, TRUE, TRUE
```

This example sets the chapter layout to single-sided, starting on the left.

---

{Button,ALink(VENTURA\_Chapter;;;,0,"Defaultoverview"),} [Related Topics](#)

## .FormatChapterGeneralGet (VENTURA)

**.FormatChapterGeneralGet** *.SingleSided=Boolean, .StartOnLeft=Boolean, .Condition=string, .VJParagraphTag=string, .VJCustomAmount=long, .HangingPunctuation=string, .StartOnEither=Boolean*

This function returns chapter properties for the active chapter.

Syntax	Description
.SingleSided	Specifies a numeric variable which is assigned the publication layout status: TRUE (-1) for single-sided and FALSE (0) for double-sided.
.StartOnLeft	Specifies a numeric variable which is assigned the publication start page: TRUE (-1) to start on the left side and FALSE (0) to start on the right side.
.Condition	Specifies a string variable which is assigned the chapter conditions.
.VJParagraphTag	Specifies a string variable which is assigned the paragraph tag that contains the interline spacing for the vertically justified publication.
.VJCustomAmount	Specifies a numeric variable which is assigned the custom interline spacing value.
.HangingPunctuation	Specifies a string variable which is assigned a string that contains the list of hanging punctuation. This string may appear as one or more characters followed by the overhang percentage in square brackets. For example, ";;;[50]!?,[100]^[75]".
.StartOnEither	Specifies a numeric variable which is assigned the chapter start page: TRUE (-1) to start on either side depending on where the previous chapter ends; otherwise, FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM SingleSided AS BOOLEAN
DIM StartOnLeft AS BOOLEAN
.FormatChapterGeneralGet, SingleSided, StartOnLeft
```

This example gets the chapter layout settings, specifying whether the publication is single or double sided, and whether it starts on the left of right page.

---

{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),} Related Topics

## .FormatChapterTypography (VENTURA)

**.FormatChapterTypography** .FirstBaseLine=*long*, .ColumnBalance=*Boolean*, .PairKerning=*Boolean*, .UseDiscHyphen=*Boolean*, .Widows=*long*, .Orphans=*long*, .VJWithinFrame=*long*, .VJMax=*long*, .VJAroundFrame=*long*, .VJMaxAtTop=*long*, .VJMaxAtBottom=*long*

This command controls character-to-character and line-to-line spacing for the entire chapter. You can override the chapter typography settings for any frame or manually inserted page using the Frame, Typography dialog box (Format menu) or the **.FormatFrameTypography** command.

Syntax	Description
.FirstBaseLine	Specifies where the first line of text begins on a page or in a frame 0 Cap Height (uppercase letters align flush with the top margin) 1 Inter-Line (move the text down by an amount equal to its inter-line spacing)
.ColumnBalance	Specifies the column balance property. Set to TRUE (-1) to enable; otherwise set to FALSE (0).
.PairKerning	Specifies whether to kern (which moves certain pairs of adjacent characters closer together). Set to TRUE (-1) to enable; otherwise set to FALSE (0).
.UseDiscHyphen	Specifies whether to have VENTURA use only discretionary hyphens (manually inserted hyphens) as opposed to automatic hyphens. This option affects only paragraphs that have automatic hyphenation enabled. Set to TRUE (-1) to enable; otherwise set to FALSE (0).
.Widows	Specifies the minimum number of lines that can be isolated at the top of the page or column. Two is a typical number, since this will prevent the last line of the paragraph from appearing by itself at the top of a page or column.
.Orphans	Specifies the minimum number of lines that can be isolated at the bottom of the page or column. Two is a typical number, since this will prevent the first line of the paragraph from appearing by itself at the bottom of a page or column.
.VJWithinFrame	Specifies the type of the vertical justification within a frame: 0 Off 1 Feathering 2 Carding
.VJMax	If <b>.VJWithinFrame</b> is set to 1 or 2, specifies the maximum amount of space allowed between elements (frames, paragraphs and tables) during vertical justification. 100% is the usual value for this setting. A value of 200%, for example, doubles all the maximum settings specified for each element. Use of this setting, therefore, allows you to control how much vertical justification takes place in a chapter, just by changing one number. Setting this value to 0 disables Vertical Justification.
.VJAroundFrame	Specifies how VENTURA is allowed to move frames as it vertically justifies pages: 0 Off 1 Moveable 2 Fixed
.VJMaxAtTop	If <b>.VJAroundFrame</b> is set to 1 or 2, specifies the maximum amount of space that VENTURA is allowed to add above frames in a chapter as it vertically justifies elements on a page, in tenths of a micron.
.VJMaxAtBottom	If <b>.VJAroundFrame</b> is set to 1 or 2, specifies the maximum amount of space that VENTURA is allowed to add below frames in a chapter as it vertically justifies elements on a page, in tenths of a micron.

### Note

**I** This command corresponds to the Typography tab on the Chapter Properties dialog box in Corel VENTURA. Click Publication, Chapter Properties, Typography.

### Example

```
M_INCH = LENGTHCONVERT (1, 7, 1)
.FormatChapterTypography 0, -1,,,,,, 2, 0.1*M_INCH, 0.1*M_INCH
```

The above example sets the chapter typography to use Cap Height and column balance. The vertical justification around frames are fixed to 0.1 inches maximum at the top and bottom of frames.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),} Related Topics

## .FormatChapterTypographyGet (VENTURA)

**.FormatChapterTypographyGet** *.FirstBaseLine=long, .ColumnBalance=Boolean, .PairKerning=Boolean, .UseDiscHyphen=Boolean, .Widows=long, .Orphans=long, .VJWithinFrame=long, .VJMax=long, .VJAroundFrame=long, .VJMaxAtTop=long, .VJMaxAtBottom=long*

This function returns current chapter's typographical settings.

Syntax	Description
.FirstBaseLine	Specifies a numeric variable which is assigned where the first line of text begins on a page or in a frame: 0 Cap Height (uppercase letters align flush with the top margin) 1 Inter-Line (moves the text down by an amount equal to its inter-line spacing)
.ColumnBalance	Specifies a numeric variable which is assigned the column balance setting: TRUE (-1) if enabled, otherwise FALSE (0).
.PairKerning	Specifies a numeric variable which is assigned the kern setting: TRUE (-1) if enabled; otherwise FALSE (0).
.UseDiscHyphen	Specifies a numeric variable which is assigned the discretionary hyphens setting: TRUE (-1) if enabled; otherwise FALSE (0).
.Widows	Specifies a numeric variable which is assigned the minimum number of lines that can be isolated at the top of the page or column. Two is a typical number, since this will prevent the last line of the paragraph from appearing by itself at the top of a page or column.
.Orphans	Specifies a numeric variable which is assigned the minimum number of lines that can be isolated at the bottom of the page or column. Two is a typical number, since this will prevent the first line of the paragraph from appearing by itself at the bottom of a page or column.
.VJWithinFrame	Specifies a numeric variable which is assigned the type of the vertical justification within a frame: 0 Off 1 Feathering 2 Carding
.VJMax	If <b>.VJWithinFrame</b> is set to 1 or 2, specifies a numeric variable which is assigned the maximum amount of space allowed between elements (frames, paragraphs and tables) during vertical justification, in tenths of a micron. 100% is the usual value for this setting. A value of 200%, for example, doubles all the maximum settings specified for each element.
.VJAroundFrame	Specifies a numeric variable which is assigned a value corresponding to how Corel VENTURA is allowed to move frames as it vertically justifies pages: 0 Off 1 Moveable 2 Fixed
.VJMaxAtTop	If <b>.VJAroundFrame</b> is set to 1 or 2, specifies a numeric variable which is assigned the maximum amount of space that Corel VENTURA is allowed to add above frames in a chapter as it vertically justifies elements on a page, in tenths of a micron.
.VJMaxAtBottom	If <b>.VJAroundFrame</b> is set to 1 or 2, specifies a numeric variable which is assigned the maximum amount of space that Corel VENTURA is allowed to add below frames in a chapter as it vertically justifies elements on a page, in tenths of a micron.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM CBalance AS BOOLEAN
DIM PKerning AS BOOLEAN
.FormatChapterTypographyGet FBaseLine&, CBalance, PKerning
```

This example returns the typographical attributes of the current chapter to the variables **FBaseLine**, **CBalance**, and **PKerning**.

---

{Button,ALink(VENTURA\_Chapter;;;,0,"Defaultoverview",)} Related Topics

## **.FormatFootnoteDefine (VENTURA)**

This command is obsolete in Corel VENTURA 8. Please see [FormatFootnoteSeparator](#) and [FormatChapterFootnote](#).

---

`{Button,ALink(VENTURA_Chapter;;;;,0,"Defaultoverview"),}` [Related Topics](#)




# .FormatFootnoteSeparator (VENTURA)

**.FormatFootnoteSeparator** .AddSeparatorLine=*Boolean*, .SpaceAbove=*long*, .LineWidth=*long*, .LineHeight=*long*

This command sets chapter footnote separator properties. Use the **.FormatChapterFootnote** command to enable footnotes in a chapter.

Syntax	Description
.AddSeparatorLine	Specifies whether to use a separator line to separate footnotes from the main text in the active chapter. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.SpaceAbove	Specifies the amount of space between the top of the footnote frame and the separator line in tenths of a micron.
.LineWidth	Specifies the width of the separator line in tenths of a micron.
.LineHeight	Specifies the thickness of the separator line in tenths of a micron.

### Note

 This command corresponds to the Footnotes tab of the Chapter Properties dialog box in Corel VENTURA. Click Publication, Chapter Properties, Footnotes tab.

### Example

```
M_INCH = LENGTHCONVERT (1, 7, 1)
M_POINT = LENGTHCONVERT (1, 3, 1)
.FormatChapterFootnote TRUE, 4,,, 1,,, TRUE, 1.25*M_INCH
.FormatFootnoteSeparator TRUE, 1.5*M_POINT, 6.5*M_INCH, 0.2*M_POINT
```

In the above example, the third line enables footnotes in the chapter, setting the maximum height of the footnote text to 1.25 inches. The fourth line sets the properties of the footnote separator line to use in the chapter .

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point and another variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),} Related Topics

## **.FormatFootnoteSeparatorGet (VENTURA)**

**.FormatFootnoteSeparatorGet** *.AddSeparatorLine=Boolean, .SpaceAbove=long, .LineWidth=long, .LineHeight=long*

This command gets the footnote separator line settings.

<b>Syntax</b>	<b>Description</b>
<code>.AddSeparatorLine</code>	Specifies a numeric variable which is assigned the separator line setting: TRUE (-1) if footnotes are separated from the main text in the active chapter; otherwise FALSE (0).
<code>.SpaceAbove</code>	Specifies a numeric variable which is assigned the amount of space between the top of the footnote frame and the separator line in tenths of a micron.
<code>.LineWidth</code>	Specifies a numeric variable which is assigned the width of the separator line in tenths of a micron.
<code>.LineHeight</code>	Specifies a numeric variable which is assigned the thickness of the separator line in tenths of a micron.

### **Example**

```
DIM AddSeparatorLine AS BOOLEAN
.FormatFootnoteSeparatorGet AddSeparatorLine, SpaceAbove&, LineWidth&, LineHeight&
```

This example returns the footnote separator attributes for the current chapter to the variables **AddSeparatorLine**, **SpaceAbove**, **LineWidth**, and **LineHeight**.

---

**{Button,ALink(VENTURA\_Chapter;;;;,0,"Defaultoverview"),}** **Related Topics**

## **.FormatUpdateAutoNumber (VENTURA)**

### **.FormatUpdateAutoNumber**

This command updates the autonumbering of the current chapter.

#### **Note**



This command corresponds to the Update Publication dialog box in Corel VENTURA. Click Publication, Update Publication.

#### **Example**

`.FormatUpdateAutoNumber`

---

`{Button,ALink(VENTURA_Chapter;;;;,0,"Defaultoverview"),}` [Related Topics](#)

## .CharacterTagAddNew (VENTURA)

**.CharacterTagAddNew .TagName=*string*, .CopyFrom=*string***

This command adds a new character tag to the current style sheet. The new tag is based on an existing character tag.

Syntax	Description
<b>.TagName</b>	Specifies the name of the new character tag. The tag name must not already exist in the style sheet.
<b>.CopyFrom</b>	Specifies the character tag that the new character tag is based on.

### Note



This command corresponds to the Tag Window. Click Tools, Tag Window.

### Example

```
.CharacterTagAddNew "keywords", "bold"
```

The above example adds a character tag named keywords to the current style sheet. The new tag is based on the bold tag.

---

{button ,AL("VENTURA\_CharacterTagAddNew\_Menu;vent\_char;;;",0,"Defaultoverview",)} Related Topics


# .CharacterTagCopy (VENTURA)

.CharacterTagCopy .TagName=*string*

This command copies a character tag from the current style sheet to the clipboard. This command can be used to copy tags into VENTURA Libraries.

Syntax	Description
.TagName	Specifies the name of the character tag to copy.

**Note**

 This command corresponds to the Tag Window. Click Tools, Tag Window.

**Example**

```
.CharacterTagCopy "TradeMark"  
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste
```

The above example copies the character tag TradeMark, creates a VENTURA library named MYLIB.VLB, and pastes the TradeMark tag into the new library.


## .CharacterTagCount (VENTURA)

**ReturnValue**& = .CharacterTagCount ( )

This function returns the number of character tags in the current style sheet.

Syntax	Description
<b>ReturnValue</b> &	Specifies the numeric variable that is assigned the number of character tags in the current style sheet.

### Note

 This command cannot be recorded.

### Example

```
Ccounts& = .CharacterTagCount ( )
```


The number of character tags is assigned the **Ccounts** variable.


---

{button ,AL("VENTURA\_CharacterTagCount\_Menu;vent\_char;;;",0,"Defaultoverview",)} [Related Topics](#)

## .CharacterTagDelete (VENTURA)


**.CharacterTagDelete .TagName=string, .ReformatTagName=string**


This command deletes a character tag from the current style sheet. Any item in the document  and in other documents that are linked to the same style sheet through the Library

 formatted with the deleted tag will be reformatted using another tag of your choice.

Syntax	Description
<b>.TagName</b>	Specifies the name of the character tag to delete.
<b>.ReformatTagName</b>	Specifies the character tag to apply to characters tagged with <b>.TagName</b> .

### Note

 This command corresponds to the Tag Window. Click Tools, Tag Window.

 You cannot delete the Bold, Italic, or Underline tags.

### Example

`.CharacterTagDelete "PartNum" , "Parts"`

The above example deletes the character tag PartNum and applies the Parts tag to characters using the PartNum tag.

---

`{button ,AL("VENTURA_CharacterTagDelete_Menu;vent_char;;;','0,"Defaultoverview"),}} Related Topics`


## .CharacterTagGetAt (VENTURA)

ReturnString\$ = .CharacterTagGetAt .TagIndex=*long*

This function returns the character tag name associated with a character tag index number in the current style sheet.

Syntax	Description
ReturnString\$	Specifies the string variable that is assigned the name of a character tag.
.TagIndex	Specifies the index number of a character tag. Index numbers are not associated with tag order in the Tag Window. Instead, index numbers are associated by the order in which the character tags are created. The first tag created is 1, the second tag created is 2, and so on. The last created tag is equal to the return value from the <a href="#">.CharacterTagCount</a> function. If a tag is deleted, the tag index is recompiled.

### Note

 This command cannot be recorded.

### Example

```
LastTag& = .CharacterTagCount ( )      'number of tags  
LastTagName$ = .CharacterTagGetAt (LastTag&)
```

In the above example, the first statement counts the number of character tags. The second statement is assigned the name of last created character tag still available in the current style sheet.

---

{button ,AL("VENTURA\_CharacterTagGetAt\_Menu;vent\_char;;;",0,"Defaultoverview",)} [Related Topics](#)



## .CharacterTagRename (VENTURA)

**.CharacterTagRename .OldTagName=string, .NewTagName=string**

This command renames a specified character tag. Renaming tags may affect the formatting of other documents that are linked to the same style sheet through the Library.

Syntax	Description
<b>.OldTagName</b>	Specifies the character tag to be renamed.
<b>.NewTagName</b>	Specifies the new name for the specified character tag.

### Note



This command corresponds to the Tag Window. Click Tools, Tag Window.



You cannot rename the Bold, Italic, or Underline tags.

### Example

```
.CharacterTagRename "Name 1", "Name 2"
```

The above example renames the character tag Name 1 to Name 2.

---

{button ,AL("VENTURA\_CharacterTagRename\_Menu;vent\_char;;;";0,"Defaultoverview",)} [Related Topics](#)

## **.FormatCharTagBegin (VENTURA)**

**.FormatCharTagBegin** *.TagName=string*

Together, the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands are required to change a character tag's attributes. The two commands enclose VENTURA commands that can modify a character tag. The **.FormatCharTagBegin** statement is placed before the first character modifying command, and the **.FormatCharTagEnd** command is placed after the last character modifying command.

Syntax	Description
<b>.TagName</b>	Specifies the paragraph tag to modify. The style must already exist in the document.

### **Note**

The following commands can be used to modify a character tag.

-  **FormatCharTagColor command**
-  **FormatCharTagDefaults command**
-  **FormatCharTagFont command**
-  **FormatCharTagHyphenation command**

### **Example**

```
.FormatCharTagBegin "Sample"  
    .FormatCharTagFont .....  
    .FormatCharTagDefaults .....  
.FormatCharTagEnd
```

In the above example, the Sample paragraph tag has its Font and Default attributes modified.


---

{button ,AL('VENTURA\_FormatCharTagBegin\_Menu;vent\_char;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FormatCharTagColor (VENTURA)

**.FormatCharTagColor** .Background=*Boolean*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .Overprint=*Boolean*

This command changes the font color and background attributes of a character tag. To change a character tag's attributes, this command must be enclosed by the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands.

Syntax	Description
.Background	Specifies whether to change attributes for the background or font colors. Set to TRUE (-1) to set background colors. Set to FALSE (0) to set font colors.
.ColorModel	<p>Specifies the color model to use for the paragraph background or font colors:</p> <ol style="list-style-type: none"><li>1 <a href="#">PANTONE Spot Colors</a></li><li>2 <a href="#">CMYK</a></li><li>3 <a href="#">CMYK255</a></li><li>4 <a href="#">CMY</a></li><li>5 <a href="#">RGB</a> (default if omitted)</li><li>6 <a href="#">HSB</a></li><li>7 <a href="#">HLS</a></li><li>8 <a href="#">Black and White</a></li><li>9 <a href="#">Grayscale</a></li><li>11 <a href="#">YIQ</a></li><li>12 <a href="#">L*a*b*</a></li><li>14 <a href="#">PANTONE Hexachrome</a></li></ol> <p>If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.</p>
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Overprint	<p>When the <b>.Background</b> parameter is set to TRUE, specifies whether to have the text's background print over objects beneath it. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0).</p> <p>When the <b>.Background</b> parameter is set to FALSE, specifies whether to have the text print over the background color. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0).</p> <p>This option is relevant only when you print color separations.</p>

### Example

```
.FormatCharTagBegin "Sample"
    .FormatCharTagColor -1, 5, 0, 30, 240
.FormatCharTagEnd
```

The above example sets the font background to a dark blue color using the RGB color model for the Sample character tag. The last parameter is omitted because RGB only uses 3 color components.

---

{button ,AL('VENTURA\_FormatCharTagColor\_Menu;vent\_char;;;',0,"Defaultoverview",)} [Related Topics](#)

## .FormatCharTagColorGet (VENTURA)

**.FormatCharTagColorGet** .Background=*Boolean*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .Overprint=*Boolean*

This function returns a character tag's font color and background attributes. To specify the character tag, enclose the function with the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands.

Syntax	Description
.Background	Specifies whether to return the background or font color attributes. Set to TRUE (-1) to return background colors. Set to FALSE (0) to return font colors. If omitted, it is set to FALSE.
.ColorModel	Specifies the numeric variable that is assigned the selected text's color mode: <div><div>1</div><div>PANTONE Spot Colors</div></div> <div><div>2</div><div>CMYK</div></div> <div><div>3</div><div>CMYK255</div></div> <div><div>4</div><div>CMY</div></div> <div><div>5</div><div>RGB (default if omitted)</div></div> <div><div>6</div><div>HSB</div></div> <div><div>7</div><div>HLS</div></div> <div><div>8</div><div>Black and White</div></div> <div><div>9</div><div>Grayscale</div></div> <div><div>11</div><div>YIQ</div></div> <div><div>12</div><div>L*a*b*</div></div> <div><div>14</div><div>PANTONE Hexachrome</div></div>
.Color1	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color3	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color4	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Overprint	Specifies the numeric variable that is assigned a value indicating whether the overprint attribute is enabled: TRUE (-1) if enabled; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatCharTagBegin "Sample"  
    .FormatCharTagColorGet -1, cm&, c1&, c2&, c3&, c4&  
.FormatCharTagEnd
```

In above example the **.FormatCharTagColorGet** function the assigns the background color settings for the Sample character tag to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

---

{button ,AL('VENTURA\_FormatCharTagColorGet\_Menu;vent\_char;;;',0,"Defaultoverview"),} Related Topics

## .FormatCharTagDefaults (VENTURA)

**.FormatCharTagDefaults** .OverscoreThickness=*long*, .OverscoreShift=*long*, .StrikeThickness=*long*, .StrikeShift=*long*, .Underline1Thickness=*long*, .Underline1Shift=*long*, .Underline2Thickness=*long*, .Underline2Shift=*long*, .SuperscriptPointSize=*single*, .SuperscriptShift=*long*, .SubscriptPointSize=*single*, .SubscriptShift=*long*, .Tracking=*long*, .TrackAmount=*long*

This command changes the default attributes of a character tag. To change a character tag's attributes, this command must be enclosed by the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands.

Syntax	Description
.OverscoreThickness	Specifies the thickness for the overscore line, in tenths of a micron.
.OverscoreShift	Specifies the distance between the overscore and the baseline of the paragraph, in tenths of a micron. A positive value moves the overscore away from the baseline; a negative value moves it closer.
.StrikeThickness	Specifies the thickness for the strike-thru line, in tenths of a micron.
.StrikeShift	Specifies the distance between the strike-thru line and the baseline of the paragraph, in tenths of a micron. A positive value moves the strike-through line away from the baseline; a negative value moves it closer.
.Underline1Thickness	Specifies the thickness for underline 1, in tenths of a micron.
.Underline1Shift	Specifies the distance between underline 1 and the baseline of the paragraph, in tenths of a micron. A positive value moves the underline away from the baseline; a negative value moves it closer.
.Underline2Thickness	Specifies the thickness for underline 2, in tenths of a micron.
.Underline2Shift	Specifies the distance between underline 2 and the baseline of the paragraph, in tenths of a micron. A positive value moves the underline away from the baseline; a negative value moves it closer.
.SuperscriptPointSize	Specifies the a font size for the superscript character, in tenths of a micron.
.SuperscriptShift	Specifies the distance between the superscript character the baseline of the paragraph, in tenths of a micron.
.SubscriptPointSize	Specifies the a font size for the subscript character, in tenths of a micron.
.SubscriptShift	Specifies the distance between the subscript character the baseline of the paragraph, in tenths of a micron.
.Tracking	Specifies the overall spacing between characters: 0 Normal 1 Tighter 2 Very tight 3 Looser 4 Very loose 5 Custom (see the <b>.TrackAmount</b> parameter)
.TrackAmount	When <b>.Tracking</b> is set to Custom, specifies the spacing between characters. This setting is specified as a percentage of an em, a unit of measure roughly equal to the width of the capital M at the current font size.

### Note

**I** This command corresponds to the Default tab on Character Tags dialog box in Corel VENTURA. Click Format, Character Tag, Default.

### Example

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatCharTagBegin "Sample"
.FormatCharTagDefaults 2*M_POINT, , , , 3*M_POINT
.FormatCharTagEnd
```

The above example sets the overscore thickness to 2 points and the **Underline 1** thickness to 3 points in the Sample character tag.

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in a point.

---

{button ,AL('VENTURA\_FormatCharTagDefaults\_Menu;vent\_char;;;',0,"Defaultoverview",)} **Related Topics**

## .FormatCharTagDefaultsGet (VENTURA)

**.FormatCharTagDefaultsGet** .OverscoreThickness=*long*, .OverscoreShift=*long*, .StrikeThickness=*long*, .StrikeShift=*long*, .Underline1Thickness=*long*, .Underline1Shift=*long*, .Underline2Thickness=*long*, .Underline2Shift=*long*, .SuperscriptPointSize=*single*, .SuperscriptShift=*long*, .SubscriptPointSize=*single*, .SubscriptShift=*long*, .Tracking=*long*, .TrackAmount=*long*

This function returns a character tag's default attributes. To specify the character tag, enclose the function with the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands.

Syntax	Description
.OverscoreThickness	Specifies the numeric variable that is assigned the thickness for the overscore line, in tenths of a micron.
.OverscoreShift	Specifies the numeric variable that is assigned the distance between the overscore and the baseline of the paragraph, in tenths of a micron.
.StrikeThickness	Specifies the numeric variable that is assigned the thickness for the strike-thru line, in tenths of a micron.
.StrikeShift	Specifies the numeric variable that is assigned the distance between the strike-thru line and the baseline of the paragraph, in tenths of a micron.
.Underline1Thickness	Specifies the numeric variable that is assigned the thickness for underline 1, in tenths of a micron.
.Underline1Shift	Specifies the numeric variable that is assigned the distance between underline 1 and the baseline of the paragraph, in tenths of a micron.
.Underline2Thickness	Specifies the numeric variable that is assigned the thickness for underline 2, in tenths of a micron.
.Underline2Shift	Specifies the numeric variable that is assigned the distance between underline 2 and the baseline of the paragraph, in tenths of a micron.
.SuperscriptPointSize	Specifies the numeric variable that is assigned the a font size for the superscript character, in tenths of a micron.
.SuperscriptShift	Specifies the numeric variable that is assigned the distance between the superscript character the baseline of the paragraph, in tenths of a micron.
.SubscriptPointSize	Specifies the numeric variable that is assigned the font size for the subscript character, in tenths of a micron.
.SubscriptShift	Specifies the numeric variable that is assigned the distance between the subscript character the baseline of the paragraph, in tenths of a micron.
.Tracking	Specifies the numeric variable that is assigned a number corresponding to the overall spacing between characters: 0 Normal 1 Tighter 2 Very tight 3 Looser 4 Very loose 5 Custom
.TrackAmount	Specifies the numeric variable that is assigned the spacing between characters as a percentage of an em, a unit of measure roughly equal to the width of the capital M at the current font size. This value is only available when <b>.Tracking</b> is set to Custom.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatCharTagBegin "Sample"  
    .FormatCharTagDefaultsGet OT&, , ST&  
.FormatCharTagEnd
```

The above example assigns the Sample character tag's overscore thickness and strike thickness to the variables **OT** and **ST**, respectively.

## **.FormatCharTagEnd (VENTURA)**

### **.FormatCharTagEnd**

Together, the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands are required to change a character tag's attributes. The two commands enclose VENTURA commands that can modify a character tag. The **.FormatCharTagBegin** statement is placed before the first character modifying command and the **.FormatCharTagEnd** command is placed after the last character modifying command. Click Related Topics for a list of VENTURA commands that can modify a character tag.

#### **Note**

The following commands can be used to modify a character tag.



**FormatCharTagColor command**



**FormatCharTagDefaults command**



**FormatCharTagFont command**



**FormatCharTagHyphenation command**

#### **Example**

```
.FormatCharTagBegin "Sample"  
    .FormatCharTagFont .....  
    .FormatCharTagDefaults .....  
.FormatCharTagEnd
```

In the above example, the Sample paragraph tag has its Font and Default attributes modified.

---

{button ,AL('VENTURA\_FormatCharTagEnd\_Menu;vent\_char;;;',0,"Defaultoverview",)} **Related Topics**

## **.FormatCharTagFont (VENTURA)**

**.FormatCharTagFont** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .ShiftUp=*long*, .Underline=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*, .Uppercase=*Boolean*, .Effect=*long*, .TextBefore=*string*, .TextAfter=*string*

This command changes the font attributes of a character tag. To change a character tag's attributes, this command must be enclosed by the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands.

Syntax	Description																		
.FontName	Specifies the name of the font. For a list of installed fonts, see the font box on the text or property toolbars.																		
.PointSize	Specifies the font size in points.																		
.Weight	<p>Specifies the font's weight setting (number of inked pixels per 1000 pixels). The weight setting ranges from 0 to 1000. Common values and their corresponding names include:</p> <table><tr><td>100</td><td>Thin</td></tr><tr><td>200</td><td>Extra Light, Ultra Light</td></tr><tr><td>300</td><td>Light</td></tr><tr><td>400</td><td>Normal, Regular</td></tr><tr><td>500</td><td>Medium</td></tr><tr><td>600</td><td>Semi Bold, Demi Bold</td></tr><tr><td>700</td><td>Bold</td></tr><tr><td>800</td><td>Extra Bold, Ultra Bold</td></tr><tr><td>900</td><td>Black, Heavy</td></tr></table> <p>This parameter is ignored if the font you're applying this setting to doesn't have the specified weight installed on your system. For example, Courier is normally available with only Normal (400) and Bold (700) weight settings. If you set the weight for a Courier font to a value other than 400 or 700, this parameter is ignored and the default setting is used.</p> <p>Some fonts may use different names than those shown above for the same weight settings. See a font dialog box for more information about the fonts you have installed on your system.</p>	100	Thin	200	Extra Light, Ultra Light	300	Light	400	Normal, Regular	500	Medium	600	Semi Bold, Demi Bold	700	Bold	800	Extra Bold, Ultra Bold	900	Black, Heavy
100	Thin																		
200	Extra Light, Ultra Light																		
300	Light																		
400	Normal, Regular																		
500	Medium																		
600	Semi Bold, Demi Bold																		
700	Bold																		
800	Extra Bold, Ultra Bold																		
900	Black, Heavy																		
.Italic	Specifies whether to apply italic formatting. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). You must have an italic version of the specified font installed on your system, or else this parameter is ignored.																		
.ShiftUp	Specifies the distance to move the selected text off the baseline. A negative number moves the text down. This value is expressed in inches multiplied by 300, rounded to the nearest whole number. To express this value using points, multiply the number of points by 300, divide by 72, and round to the nearest whole number.																		
.Underline	<p>Specifies the font's underline attributes:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Single</td></tr><tr><td>2</td><td>Double</td></tr><tr><td>3</td><td>Word underline</td></tr></table>	0	None	1	Single	2	Double	3	Word underline										
0	None																		
1	Single																		
2	Double																		
3	Word underline																		
.Strikethru	Specifies the font's strike-thru setting. Set to TRUE (-1) to enable strike-thru formatting. Set to FALSE (0) to disable strike-thru formatting.																		
.Overscore	Specifies the font's overscore setting. Set to TRUE (-1) to enable overscore formatting. Set to FALSE (0) to disable italics formatting.																		
.Uppercase	Specifies whether to use uppercase characters with the font. Set to TRUE (-1) to enable uppercase characters; otherwise, set to FALSE (0).																		
.Effect	<p>Specifies special effects setting:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Subscript</td></tr><tr><td>2</td><td>Superscript</td></tr><tr><td>3</td><td>Small caps</td></tr></table>	0	None	1	Subscript	2	Superscript	3	Small caps										
0	None																		
1	Subscript																		
2	Superscript																		
3	Small caps																		
.TextBefore	Specifies the identifier of a previously defined Text Before/After text string. The text string is inserted before the characters formatted with the current tag. To create a text string, the <b>.FormatPubTextAdd</b> command.																		
.TextAfter	Specifies the identifier of a previously defined Text Before/After text string. The text string is inserted after the characters formatted with the current tag. To create a text string, the <b>.FormatPubTextAdd</b> command.																		

### **Note**



This command corresponds to the Font tab on Character Tags dialog box in Corel VENTURA. Click Format, Character



Tag, Font.

**Example**

```
.FormatCharTagBegin "Sample"  
    .FormatCharTagFont "AvantGarde", 12, 700, , , , -1  
.FormatCharTagEnd
```

The above example sets the font to AvantGarde, 12 points, boldface, and uppercase characters.

---

{button ,AL('VENTURA\_FormatCharTagFont\_Menu;vent\_char;;;','0,"Defaultoverview",,)} [Related Topics](#)

## .FormatCharTagFontGet (VENTURA)

**.FormatCharTagFontGet** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .Underline=*long*, .ShiftUp=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*, .Uppercase=*Boolean*, .Effect=*long*, .TextBefore=*string*, .TextAfter=*string*

This function returns a character tag's font attributes. To specify the character tag, enclose the function with the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands.

Syntax	Description
.FontName	Specifies a string variable that is assigned the name of the font.
.PointSize	Specifies a numeric variable that is assigned the font size in points.
.Weight	Specifies a numeric variable that is assigned the font's weight setting (number of inked pixels per 1000 pixels). The weight setting ranges from 0 to 1000. Common values and their corresponding names include: 100 Thin 200 Extra Light, Ultra Light 300 Light 400 Normal, Regular 500 Medium 600 Semi Bold, Demi Bold 700 Bold 800 Extra Bold, Ultra Bold 900 Black, Heavy
.Italic	Specifies a numeric variable that is assigned the font's italic setting: TRUE (-1) if enabled; FALSE (0) otherwise.
.ShiftUp	Specifies the numeric variable that is assigned a return value corresponding to the selected text's baseline shift. This value is expressed in inches multiplied by 300, rounded to the nearest whole number.
.Underline	Specifies a numeric variable that is assigned the font's underline attributes: 0 None 1 Single 2 Double 3 Word underline
.StrikeThru	Specifies a numeric variable that is assigned the font's strike-thru setting: TRUE (-1) if strike-thru formatting is enabled; FALSE (0) otherwise.
.Overscore	Specifies a numeric variable that is assigned the font's overscore setting: TRUE (-1) if overscore formatting is enabled; FALSE (0) otherwise.
.Uppercase	Specifies a numeric variable that is assigned a value indicating whether to use uppercase characters with the font: TRUE (-1) if uppercase characters is enabled; FALSE (0) otherwise.
.Effect	Specifies a numeric variable that is assigned the special effects setting: 0 None 1 Subscript 2 Superscript 3 Small caps
.TextBefore	Specifies a string variable that is assigned the identifier of a previously defined Text Before/After text string (if exists) to insert before the characters formatted with the current tag. To create a text string, the <b>.FormatPubTextAdd</b> command.
.TextAfter	Specifies a string variable that is assigned the identifier of a previously defined Text Before/After text string (if exists) to insert after the characters formatted with the current tag. To create a text string, the <b>.FormatPubTextAdd</b> command.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatCharTagBegin "Sample"  
    DIM Iset AS BOOLEAN  
    .FormatCharTagFontGet Fname$, Fpoint!, ,Iset  
.FormatCharTagEnd
```

The above example assigns the Sample character tag's font name, point size, and italic setting to the variables **Fname**, **Fpoint**, and **Iset**, respectively.

---

{button ,AL(^VENTURA\_FormatCharTagFontGet\_Menu;vent\_char;;;',0,"Defaultoverview"),}} [Related Topics](#)

## .FormatCharTagHyphenation (VENTURA)

**.FormatCharTagHyphenation** .Automatic=*Boolean*, .Dictionary=*string*, .MinWordSize=*long*, .MinCharactersBefore=*long*, .MinCharactersAfter=*long*, .ForCapitalized=*Boolean*

This command changes the hyphenation attributes of a character tag. To change a character tag's attributes, this command must be enclosed by the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands.

Syntax	Description
.Automatic	Specifies whether to automatically hyphenate text using the dictionary specified in the <b>.Dictionary</b> parameter. Set to TRUE (-1) to enable this option, otherwise, set to FALSE (0).
.Dictionary	Specifies the language dictionary VENTURA uses when hyphenating.
.MinWordSize	Specifies the minimum length of hyphenated words. For example, if you do not want words of less than five characters hyphenated, set to four. The <b>.Automatic</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.MinCharactersBefore	Specifies the minimum number of characters that can precede a hyphen. The <b>.Automatic</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.MinCharactersAfter	Specifies the minimum number of characters that can follow a hyphen. The <b>.Automatic</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.ForCapitalized	Specifies whether to prevent hyphenation of uppercase words such as a product name. Set to TRUE (-1) to prevent hyphenation of uppercase words; otherwise, set to FALSE (0). The <b>.Automatic</b> parameter must be set to TRUE (-1), or else this parameter is ignored.

### Note

**I** This command corresponds to the Hyphenation tab of the Character Tag dialog box in Corel VENTURA. Click Format, Character Tag, Hyphens.

### Example

```
.FormatCharTagBegin "Sample"  
    .FormatCharTagHyphenation -1, "Italian", , 4  
.FormatCharTagEnd
```

The above example enables automatic hyphenation using the Italian dictionary and sets the minimum word to hyphenate to 4 characters in the Sample character tag.

---

{button ,AL('VENTURA\_FormatCharTagHyphenation\_Menu;vent\_char;;;',0,"Defaultoverview"),} [Related Topics](#)



# .FormatCharTagHyphenationGet (VENTURA)

**.FormatCharTagHyphenationGet** *.Automatic=Boolean, .Dictionary=string, .MinWordSize=long, .MinCharactersBefore=long, .MinCharactersAfter=long, .ForCapitalized=Boolean*

This function returns a character tag's hyphenation attributes. To specify the character tag, enclose the function with the **.FormatCharTagBegin** and **.FormatCharTagEnd** commands.

Syntax	Description
.Automatic	Specifies the numeric variable that is assigned the automatic hyphenation attribute: TRUE (-1) if this option is enabled; FALSE (0) otherwise.
.Dictionary	Specifies the numeric variable that is assigned the language dictionary VENTURA uses when hyphenating the paragraph.
.MinWordSize	Specifies the numeric variable that is assigned the minimum length of hyphenated words in the paragraph.
.MinCharactersBefore	Specifies the numeric variable that is assigned the minimum number of characters that can precede a hyphen.
.MinCharactersAfter	Specifies the numeric variable that is assigned the minimum number of characters that can follow a hyphen.
.ForCapitalized	Specifies the numeric variable that is assigned the a value corresponding to whether hyphenation of capitalized words is prevented: TRUE (-1) if hyphenation is prevented; FALSE (0) otherwise.

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

## Example

```
.FormatCharTagBegin "Sample"  
    DIM AH AS BOOLEAN  
    .FormatCharTagHyphenationGet AH, Dic$, , MinWord&  
.FormatCharTagEnd
```

The above example assigns the automatic hyphenation, dictionary, and minimum word size for the Sample character tag to the variables **AH**, **Dic**, and **MinWord**, respectively.

## **.FormatSetCharacterTag (VENTURA)**

**.FormatSetCharacterTag** *.TagName=string*

This command applies a specified character tag to selected text.

Syntax	Description
<b>.TagName</b>	Specifies the name of an existing character tag to apply.

### **Note**



This command corresponds to selecting text and choosing a character tag from the Character Tag list box in the Corel VENTURA property bar.

### **Example**

```
.TextStartOfLine TRUE  
.FormatSetCharacterTag "TradeMark"
```

The above example selects the text on the current line from the beginning of the line to the insertion point. The selected text then has the TradeMark character tag applied to it.

---

{button ,AL('VENTURA\_FormatCharTagHyphenationGet\_Menu;vent\_char;;;','0',"Defaultoverview"),} Related Topics

<b>Color Model:</b>	PANTONE Spot Colors
<b>Model Number:</b>	1
<b>Color1:</b>	Pantone ID Number
<b>Color2:</b>	Tint (0 - 100)
<b>Color3:</b>	Not used
<b>Color4:</b>	Not used

<b>Color Model:</b>	CMYK
<b>Model Number:</b>	2
<b>Color1:</b>	Cyan (0 - 100)
<b>Color2:</b>	Magenta (0 - 100)
<b>Color3:</b>	Yellow (0 - 100)
<b>Color4:</b>	Black (0 - 100)



**Color Model:** CMYK255

**Model Number:** 3

**Color1:** Cyan (0 - 255)

**Color2:** Magenta (0 - 255)

**Color3:** Yellow (0 - 255)

**Color4:** Black (0 - 255)

<b>Color Model:</b>	CMY
<b>Model Number:</b>	4
<b>Color1:</b>	Cyan (0 - 255)
<b>Color2:</b>	Magenta (0 - 255)
<b>Color3:</b>	Yellow (0 - 255)
<b>Color4:</b>	Not used

<b>Color Model:</b>	RGB
<b>Model Number:</b>	5
<b>Color1:</b>	Red (0 - 255)
<b>Color2:</b>	Green (0 - 255)
<b>Color3:</b>	Blue (0 - 255)
<b>Color4:</b>	Not used

<b>Color Model:</b>	HSB
<b>Model Number:</b>	6
<b>Color1:</b>	Hue (0 - 360)
<b>Color2:</b>	Saturation (0 - 255)
<b>Color3:</b>	Brightness (0 - 255)
<b>Color4:</b>	Not used

<b>Color Model:</b>	HLS
<b>Model Number:</b>	7
<b>Color1:</b>	Hue (0 - 360)
<b>Color2:</b>	Lightness (0 - 255)
<b>Color3:</b>	Saturation (0 - 255)
<b>Color4:</b>	Not used

<b>Color Model:</b>	Black and White
<b>Model Number:</b>	8
<b>Color1:</b>	0 (Black) or 1 (White)
<b>Color2:</b>	Not used
<b>Color3:</b>	Not used
<b>Color4:</b>	Not used

<b>Color Model:</b>	Grayscale
<b>Model Number:</b>	9
<b>Color1:</b>	Black (0-255)
<b>Color2:</b>	Not used
<b>Color3:</b>	Not used
<b>Color4:</b>	Not used

<b>Color Model:</b>	YIQ
<b>Model Number:</b>	11
<b>Color1:</b>	Y-luminance (0 - 255)
<b>Color2:</b>	I-chromaticity (0 - 255)
<b>Color3:</b>	Q-chromaticity (0 - 255)
<b>Color4:</b>	Not used



**Color Model:** L\*a\*b\*  
**Model Number:** 12  
**Color1:** L\*-lightness (0 - 255)  
**Color2:** a\*-green to red (-127 - 127)  
**Color3:** b\*-blue to yellow (-127 - 127)  
**Color4:** Not used

The values specified above do not correspond to the values used in a VENTURA color dialog box. If you record a script that uses L\*a\*b\* settings, the color dialog box values are converted. The **L\*-lightness** parameter is multiplied by 2.55 and is rounded to the nearest whole number. The **a\*** and **b\*** parameters are multiplied by 2.117 and are rounded to the nearest whole number.

<b>Color Model:</b>	PANTONE Hexachrome
<b>Model Number:</b>	14
<b>Color1:</b>	Pantone ID Number
<b>Color2:</b>	Tint (0 - 100)
<b>Color3:</b>	Not used
<b>Color4:</b>	Not used

Palette	Color Model
Uniform Colors	RGB
FOCOLTONE Colors	CMYK
PANTONE Process Colors	CMYK
TRUMATCH Colors	CMYK
SpectraMaster Colors	CMYK
TOYO COLOR FINDER	CMYK
DIC Colors	CMYK

## VENTURA Text Import filters

Filter ID	Filter
2	Microsoft Word for Windows 6.0, 7.0 (.DOC)
3	Microsoft Word for Windows 1.x (.DOC)
4	Microsoft Word for Windows 2.x (.DOC)
5	Microsoft Word 3.x (.DOC)
6	Microsoft Word 4.x (.DOC)
7	Microsoft Word 5.0, 5.5 (.DOC)
8	Microsoft Word for Macintosh 3.0 (.DOC)
9	Microsoft Word for Macintosh 4.0 (.DOC)
10	Microsoft Word for Macintosh 5.0 (.DOC)
11	Rich Text Format (.RTF)
12	Corel WordPerfect 6.x (.WP*)
13	Corel WordPerfect 5.1, 5.2 (.WP*)
14	Corel WordPerfect 5.0 (.WP*)
15	Corel WordPerfect 4.2 (.WP*)
16	WordStar for Windows 1.x (.WSD)
17	WordStar for Windows 2.0 (.WSD)
18	WordStar 7.0 (.WSD)
19	WordStar 6.0 (.WSD)
20	WordStar 5.0 (.WSD)
21	WordStar 4.0 (.WSD)
22	WordStar 3.45 (.WSD)
23	WordStar 3.3, 3.31 (.WSD)
24	WordStar 2000 (.WSD)
25	XYWrite III (.XY)
26	XYWrite III Plus (.XY)
27	XYWrite IV, XYWrite for Windows 1.0 (.XY)
29	Ami Professional 1.1, 1.2 (.SAM)
30	Ami Professional 2.0, 3.0
31	Legacy 1.0, 2.0 (.LEG)
32	Microsoft Excel 1.0, 3.0, 4.0 (.XLS)
33	Microsoft Excel 5.0 (.XLS)
34	Lotus 1-2-3 (.WK)
35	Corel Quattro Pro (.WB, .WQ)
36	Corel WordPerfect 7.0 (.WP*)
38	Corel WordPerfect for Macintosh 3.x (.WPM)
37	Corel WordPerfect for Macintosh 2.x (.WPM)
39	Corel WordPerfect for Macintosh 1.x (.WPM)
40	Adobe PageMaker 6.0 (.PM6)
41	Adobe FrameMaker 3.0 (MIF)
42	Adobe FrameMaker 4.0 (MIF)

43	Adobe FrameMaker 5.0 (.MIF)
50	Ventura Generated Files (.GEN)
51	ANSI text (.TXT)
52	ASCII text (.TXT)
53	ASCII 8-bit text (.TXT)
54	Lotus/Excel Print Table (.PRN)
55	TagWrite - Style Match RTF (.RTF)
56	TagWrite - SGML (.RTF)

---

{button ,AL(^ventura\_fileimporttext;;;;;0,"Defaultoverview",)} Related Topics

## Corel VENTURA Picture Export filters

Filter String	Filter
AI2	Adobe Illustrator (AI)
PSD	Adobe Photoshop (PSD)
PDF	Adobe Portable Document File (PDF)
DXF	AutoCAD (DXF)
CALS	CALS Compressed Bitmap (CAL)
GIF	CompuServe Bitmap (GIF)
CGM	Computer Graphics Metafile (CGM)
CPT6	Corel PHOTO-PAINT 6 (CPT)
CPT	Corel PHOTO-PAINT 7 and 8 (CPT)
CMX60	Corel Presentation Exchange (CMX)
CMX50	Corel Presentation Exchange 5 (CMX)
WPG	Corel WordPerfect Graphic (WPG)
DCS	Desktop Color Separation (DCS)
EPS	Encapsulated PostScript (EPS)
EMF	Enhanced Windows Metafile (EMF)
FMV	Frame Vector Metafile (FMV)
GEM	GEM File (GEM)
IMG	GEM Paint File (IMG)
HPGL	HPGL Plotter File (PLT)
PIF	IBM PIF (PF)
JPEG	JPEG Bitmap (JPG)
FPX	Kodak FlashPix Image (FPX)
PICT	Macintosh PICT (PCT)
MAC	MACPaint Bitmap (MAC)
DRW	Micrografx 2.x, 3.x (DRW)
OS2	OS/2 Bitmap (BMP)
PCX	PaintBrush (PCX)
PNG	Portable Network Graphics (PNG)
SCT	SCITEX CT Bitmap (SCT)
SCD	ScodI (SCD)
TGA	Targa Bitmap (TGA)
TIFF	Tagged Image File Format (TIF)
WVL	Wavelet Compressed Bitmap (WI)
BMP	Windows Bitmap (BMP)
WMF	Windows Metafile (WMF)

---

{button ,AL('ventura\_fileimportpicture;;;;','0,"Defaultoverview",)} [Related Topics](#)

## OLE object's server names

OLE Object	OLE object's server name
Corel PHOTO-PAINT 6	CorelPhotoPaint.Image.6
Corel Presentations 7 Chart	WPDDraw30.Drawing
Corel Quattro Pro 7 Chart	QuattroPro.Chart.7
Corel Quattro Pro 7 Notebook	QuattroPro.Notebook.7
Corel WordPerfect 7 Document	WP7Doc
Corel WordPerfect 7 Drawing	WPDDraw30.Drawing
CorelDRAW 5	CDraw5
CorelMEMO	CorelMEMO.6
Microsoft Excel Chart	Excel.Chart.5
Microsoft Excel Graph 5.0	MSGraph.Chart.5
Microsoft Excel Worksheet	Excel.Sheet.5
Microsoft Word Document	Word.Document.6
Wave Sound	SoundRec
Windows Paintbrush	Paint.Picture

---

{button ,AL('ventura\_insertobject;ventura\_EditObjectInfoGet;;;',0,"Defaultoverview"),} [Related Topics](#)

**Bitmap picture formats**

Windows Bitmap (.BMP)

Paintbrush (.PCX)

Targa Bitmap (.TGA)

TIFF Bitmap (.TIF)

CompuServe bitmap (.GIF)

JPEG Bitmap (.JPG)

Kodak Photo-CD Image (.PCD)

Scitex CT Bitmap (.SCT)

Windows 3.x/NT Cursor Resource(.CUR)

Windows 3.x/NT Icon Resource (.ICO)

Windows 3.x/NT Bitmap Resource Bitmap (.EXE)

GEM Paint File (.IMG)

Adobe PhotoShop (.PSD)

Picture Publisher 4 (.PP4)

MACPaint Bitmap (.MAC)

OS/2 Bitmap (.BMP)

WaveLet Compressed Bitmap (.WVL)

Corel PHOTO-PAINT Image (.CPT)



**Vector picture formats**

CALS Compressed Bitmap (.CAL)  
Desktop Color Separation (.DCS)  
Computer Graphics Metafile (.CGM)  
HPGL Plotter (.PLT)  
Micrografx 2.x, 3.x (.DRW)  
Adobe Illustrator (.AI, .EPS)  
GEM file (.GEM)  
IBM PIF (.PF)  
Lotus Pic (.PIC)  
Corel WordPerfect Graphic (.WPG)  
Macintosh PICT (.PCT)  
Encapsulated PostScript (.EPS)  
PostScript Interpreted (.PS)  
MET MetaFile (.MET)  
Windows Metafile (.WMF)  
Corel Metafile (.CMF)  
AutoCAD (.DXF)  
Enhanced Windows Metafile (.EMF)  
Corel TRACE (.AI, .EPS)  
CorelMOVE (.CMV, .MLB)  
CorelCHART (.CCH)  
Corel Presentation Exchange 6.0 (.CMX)  
Corel Presentation Exchange 5.0 (.CMX)  
CorelDRAW (.CDR)  
CorelDRAW Compressed (.CDX)  
Corel CMX Compressed (.CPX)

## Date and Time formats

The way that a time or date inserted using the Insert Item command (Insert menu) appears in a document depends on the date and time format you select. You can choose one of the built-in formats or create your own by typing the following codes.

### Date codes

To display	Type this format code
Days as 1-31	d
Days as 01-31	dd
Days as Sun-Sat	ddd
Days as Sunday-Saturday	dddd
Days as ordinals(e.g, 13th, 3rd)	dt
week of the year, numerical, 1 - 54	w
week of the year, numerical, 01 - 54	ww
Months as 1-12	M
Months as 01-12	MM
Months as Jan-Dec	MMM (uppercase)
Months as January-December	MMMM (uppercase)
Day of the year, 1-366	y
Years as 00-99	yy
Day of the year, 001-366	yyy
Years as 1900-2078	yyyy
Days, month or years as Roman numeral	Type R or r after the day, month or year code
Quarters as 1, 2, 3 or 4	q
1st, 2nd, 3rd or 4th quarter	qt 'quarter'

### Time codes

To display	Type this format code
Hours as 1-12 (12-hour clock)	h
Hours as 01-12 (12-hour clock)	hh
Hours as 0-23 (24-hour clock)	H
Hours as 00-23 (24-hour clock)	HH
Minutes as 0-59	m
Minutes as 00-59	mm
Seconds as 0-59	s
Seconds as 00-59	ss
AM/PM as a or p	a
AM/PM as A or P	A
Time as 4:36 pm	h:mm pm

### Note

The above commands can be suffixed with the following:

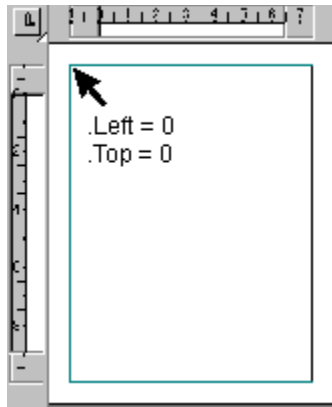
r print the number in lower-case roman numerals, i - mmmcmxcix

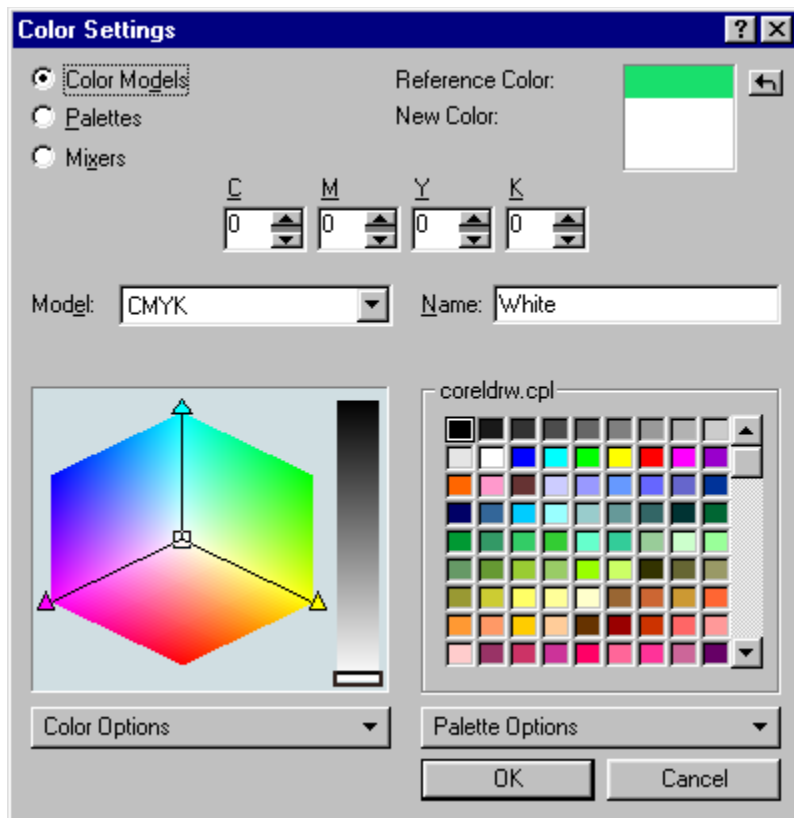
R print the number in upper-case roman numerals, I - MMMCMXCIX

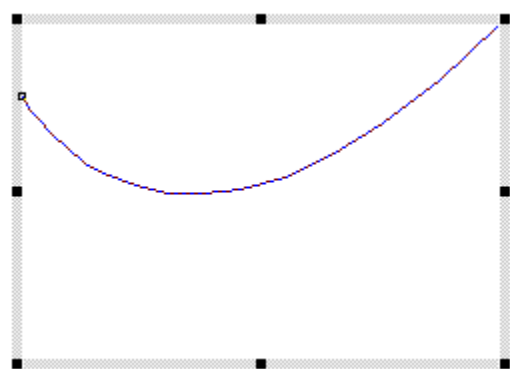
t print as ordinal, for example,. 1st, 2nd, 3rd , and so on.

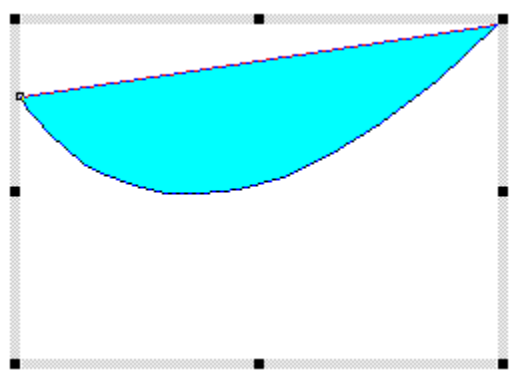
## VENTURA Paper Types

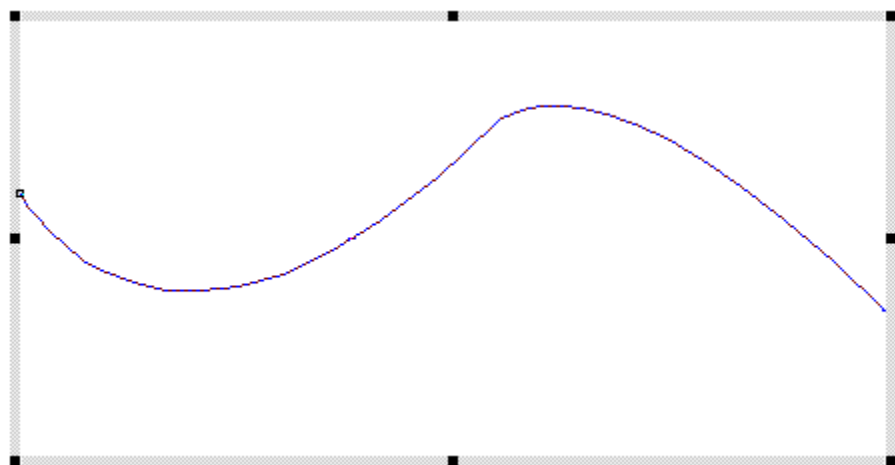
Paper ID	Paper Types
0	Letter
1	Legal
2	Tabloid
3	Statement/Half
4	Executive
5	Fan fold
6	Double
7	Broad sheet
8	A2
9	A3
10	A4
11	A5
12	A6
13	B4
14	B5
15	C3
16	C4
17	C5
18	C6
19	RA2
20	RA3
21	RA4
22	SRA3
23	SRA4
24	Envelope #9
25	Envelope #10
26	Envelope #11
27	Envelope #12
28	Envelope #14
29	Envelope monarch
30	Envelope check
31	DL
32	German fan fold
33	German legal fan fold
34	Custom



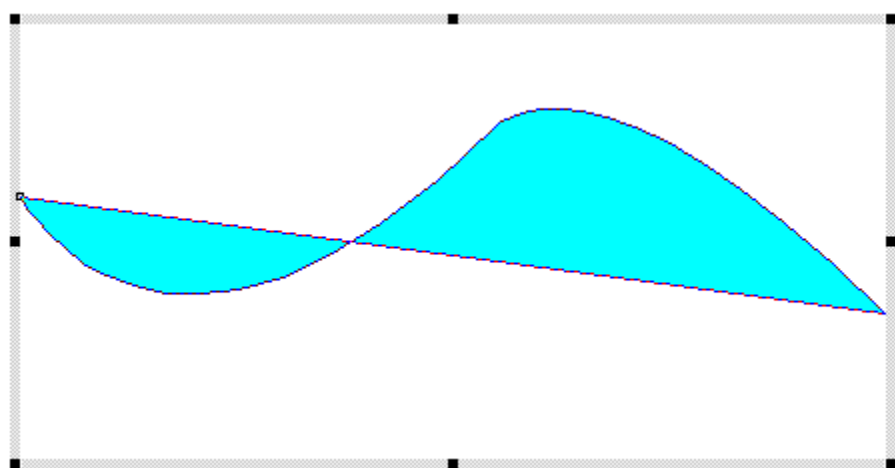


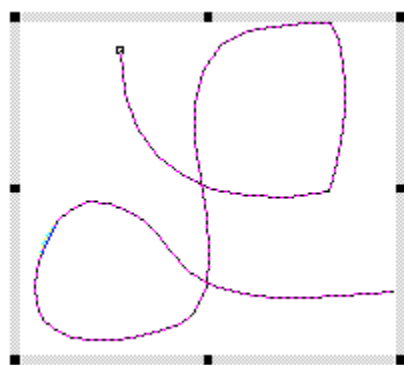


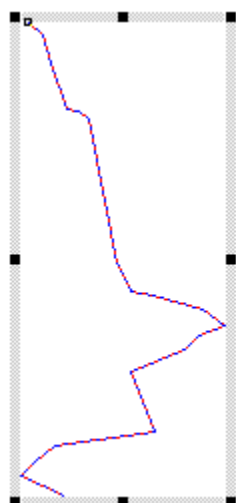


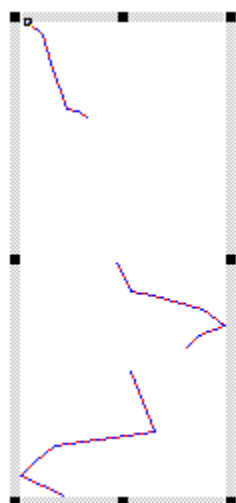













### Explicit declaration

To explicitly declare a variable, use the Corel SCRIPT [DIM](#), [GLOBAL](#), or [STATIC](#) statements. For example, to declare the variable **Fname** as a variable of data type string, the following syntax could be used:


```
DIM Fname AS STRING
```

For more information about explicit declaration, click .

### Implicit declaration

To implicitly declare a variable, use a type declaration suffix. For example, to declare **Fname** as a variable of data type string, the following syntax could be used within a VENTURA scripting command:

```
Fname$
```

For more information about implicit declaration, click .

<b>Special Item</b>	<b>Code</b>
Box Character	<\$B>
Cross-Reference	<\$R <b>required marker</b> >
Equation	<\$E>
Footnote	<\$F <b>optional marker</b> >
Endnote	<\$N <b>optional marker</b> >
Frame Anchor	<\$& <b>required marker</b> >
Index Entry	<\$I>
Marker Name	<\$M[ <b>required marker</b> ]>
Overrides	<\$T>
Hidden Text	<\$! <b>optional marker</b> >
Date/Time	<\$X>
Variable	<\$V[ <b>optional marker</b> ]>

<b>Special Character</b>	<b>Code</b>
Paragraph Return	<P>
Tab	<T>
Em Space	<_>
En Space	<->
Non-braking space	<N>
Thin Space	< >
Figure Space	<+>
Line Break	<R>
Discretionary Hyphen	<->
Open Double Quote	<@147>
Close Double Quote	<@148>
Copyright	<@169>
Registered	<@174>
Trademark	<@153>
En Dash	<@150>
Em Dash	<@151>
Any Text (Find only)	*
Any Character (Find only)	?
Clipboard Contents (Replace only)	<#>
Found Text (Replace only)	<%>



## .ColorDialog (VENTURA)

**Statement:** `.ColorDialog .WndHandle=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long`

**Function:** `ReturnValue`

**=** `.ColorDialog .WndHandle=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long`

This statement and function displays the Corel VENTURA Color Settings dialog box and returns a user's color model and color setting selections in the dialog box. On its own, the Color Settings dialog box cannot be used to change object or text colors in VENTURA. The Corel SCRIPT GETCOLOR function can also be used to display a standard Windows Color dialog box and return color setting values from the RGB color model (Red, Green, Blue).

Syntax	Description
ReturnValue	Specifies how the dialog box was closed: TRUE (-1) if the OK button was clicked; FALSE (0) if the Cancel or Close Dialog button (X) was clicked. If FALSE, the default color settings are returned in the color parameters.
.WndHandle	Specifies the numeric variable that is assigned the Windows handle of the application calling the <b>.ColorDialog</b> command. See the Corel SCRIPT command <u>GETWINHANDLE</u> for more information about Windows handles.
.ColorModel	Specifies the numeric variable that is assigned a return value corresponding to the color model the user selected: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u>
.Color1	Specifies the numeric variable that is assigned a return value corresponding to the value the user selected for the first color component of the selected color model. For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that is assigned a return value corresponding to the value the user selected for the second color component of the selected color model. For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the selected color model, 0 is returned.
.Color3	Specifies the numeric variable that is assigned a return value corresponding to the value the user selected for the third color component of the selected color model. For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the selected color model, 0 is returned.
.Color4	Specifies the numeric variable that is assigned a return value corresponding to the value the user selected for the fourth color component of the selected color model. For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the selected color model, 0 is returned.

### Note

**I** The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

**I** Assigning a value to one of the last five parameter variables before the **.ColorDialog** command is issued sets a default value for the parameter option in the Color Settings dialog box. See the example for more information.

### Example

```
cm& = 3
c1& = 235
c2& = 225
c3& = 15
c4& = 25
.ColorDialog , cm, c1, c2, c3, c4
```

The first 5 lines in the above example declare and set default values for the Color Settings dialog box. The default color model is CMYK255 (cm=3), and the color settings indicate a dark blue color. The last line could also be specified as a function as follows:

```
RV = .ColorDialog cm, c1, c2, c3, c4
```

---

{button ,AL('Ventrura\_ColorDialog\_Menu;vent\_color\_model;;;','0,"Defaultoverview",,)} [Related Topics](#)

## .ConvertColorToRGB (VENTURA)

**.ConvertColorToRGB** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This command converts VENTURA color model settings to the RGB (Red, Green, Blue) color model.

Syntax	Description
.ColorModel	Specifies the numeric variable that holds the color model value that is being converted to RGB. This variable is assigned 5 (RGB) when the <b>.ConvertColorToRGB</b> command is issued: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
.Color1	Specifies the numeric variable that holds the first color component of the selected color model. For example, Hue is the first color component for HSB. This variable is assigned a corresponding Red value when the <b>.ConvertColorToRGB</b> command is issued. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that holds the second color component of the selected color model. For example, Saturation is the second color component for HSB. This variable is assigned a corresponding Green value when the <b>.ConvertColorToRGB</b> command is issued. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the selected color model, omit it.
.Color3	Specifies the numeric variable that holds the third color component of the selected color model. For example, Saturation is the third color component for HLS. This variable is assigned a corresponding Blue value when the <b>.ConvertColorToRGB</b> command is issued. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the selected color model, omit it.
.Color4	Specifies the numeric variable that holds the fourth color component of the selected color model. For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the selected color model, omit it. This variable is not assigned a corresponding RGB value since RGB only uses three parameters.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example

```
cm& = 3
c1& = 235
c2& = 225
c3& = 15
c4& = 25
.ColorDialog cm, c1, c2, c3, c4
```

The first 5 lines in the above example declare and set values for the CMYK255 (cm=3) color model. The parameter settings indicate a dark blue color. This command returns the following values:

```
cm& = 5
c1& = 10
c2& = 20
c3& = 230
```

---

{button ,AL("Ventura\_ConvertColorToRGB\_Menu;vent\_color\_model;;;','0,"Defaultoverview"),} [Related Topics](#)

## .CAOInstall (VENTURA)


**Command:** .CAOInstall .CAOName=*string*


**Function:** ReturnValue& = .CAOInstall (.CAOName)

This command and function installs a [Corel Add-on](#) (.CAO) into VENTURA. Before you can use the functions in an installed Add-on, you must register them. See the [CAORegister](#) command for more information.

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a value corresponding to whether the specified Add-on was installed: TRUE (-1) if not installed; otherwise FALSE (0).
.CAOName	Specifies the name and path of the Add-on to install.

### Note

 This command corresponds to the Corel Add-ons dialog box in Corel VENTURA. Click Tools, Add-ons, New, and choose an Add-on.

 To uninstall a Corel Add-on, see the [CAOUninstall](#) command.

### Example

```
.CAOInstall "C:\VENTURA\CarSpec.cao"  
.CAORegister "CarSpec", "TechWordCount", "Counts automotive words"
```

In the above example, the first line installs the Corel Add-on CARSPEC.CAO. The second line registers the function.

---

{button ,AL('VENTURA\_CustomGetMenuAt\_Menu;vent\_customize;;;','0,"Defaultoverview",)} [Related Topics](#)

## .CAOInvoke (VENTURA)


**Command:** .CAOInvoke .CAOName=*string*, .FunctionName=*string*

**Function:** ReturnValue = .CAOInvoke (.CAOName, .FunctionName)

This command and function executes an installed [Corel Add-on](#) function. This command is replaces [.CAORun](#).

Syntax	Description
<b>ReturnValue</b>	Specifies the numeric variable that is assigned a value corresponding to whether the specified Add-on function was found: TRUE (-1) if not found; otherwise FALSE (0).
<b>.CAOName</b>	Specifies the name of the Add-on in which the function resides.
<b>.FunctionName</b>	Specifies the function in the specified Add-on to execute.

### Note

 This command corresponds to the Corel Add-ons dialog box in Corel VENTURA. Click Tools, Add-ons, select an Add-on function, then click Run.

### Example

```
.CAOInvoke "CarSpec", "TechWordCount"
```

The above example attempts to execute the TechWordCount function in the CARSPEC Corel Add-on. This line can also be specified as follows:

```
ReturnValue = .CAOInvoke ("CarSpec", "TechWordCount")
```

---

{button ,AL('VENTURA\_CustomGetMenuAt\_Menu;vent\_customize;;;',0,"Defaultoverview",)} [Related Topics](#)

## .CAORegister (VENTURA)

**Command:** .CAORegister .CAOName=*string*, .FunctionName=*string*, .Description=*string*

**Function:** ReturnValue = .CAORegister (.CAOName, .FunctionName, .Description)

This command and function register the functions in a previously installed [Corel Add-on](#) (.CAO). The functions must be registered before you can use them. To install a Corel Add-on use the [.CAOInstall](#) command.

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a value corresponding to whether the specified function was registered: TRUE (-1) if not registered; otherwise FALSE (0).
.CAOName	Specifies the name of the Add-on in which the function resides.
.FunctionName	Specifies the function in the specified Add-on to register.
.Description	Specifies the description to associate to the registered function. This description is displayed in a Tooltip pop-up window when the mouse pointer remains over the function name.

### Note



Each function in a Corel Add-on requires a separate registration command.

### Example

```
.CAOInstall "C:\VENTURA\CarSpec.cao"  
.CAORegister "CarSpec", "TechWordCount", "Counts automotive words"
```

In the above example, the first line installs the Corel Add-on CARSPEC.CAO. The second line registers the TechWordCount function and provides a description for it. The second line can also be specified as follows:

```
ReturnValue = .CAORegister("CarSpec", "TechWordCount", "Counts automotive words")
```

---

{button ,AL(^VENTURA\_CustomGetMenuAt\_Menu;vent\_customize;;;,0,"Defaultoverview",)} [Related Topics](#)

## .CAORun (VENTURA)


**Command:** .CAORun .CAOName=*string*, .FunctionName=*string*

**Function:** ReturnValue = .CAORun (.CAOName, .FunctionName)

This command and function executes an installed [Corel Add-on](#) function. This command is obsolete in Corel VENTURA 8. See [.CAOInvoke](#) for more information.

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a value corresponding to whether the specified Add-on function was found: TRUE (-1) if not found; otherwise FALSE (0).
.CAOName	Specifies the name of the Add-on in which the function resides.
.FunctionName	Specifies the function in the specified Add-on to execute.

### Note

 This command corresponds to the Corel Add-ons dialog box in Corel VENTURA. Click Tools, Add-ons, select an Add-on function, then click Run.

### Example

```
.CAORun "CarSpec", "TechWordCount"
```

The above example attempts to execute the TechWordCount function in the CARSPEC Corel Add-on. This line can also be specified as follows:

```
ReturnValue = .CAORun ("CarSpec", "TechWordCount")
```

---

{button ,AL("VENTURA\_CustomGetMenuAt\_Menu;vent\_customize;;;",0,"Defaultoverview",)} [Related Topics](#)

## .CAOUninstall (VENTURA)

**Command:** .CAOUninstall .CAOName=*string*

**Function:** ReturnValue = .CAOUninstall (.CAOName)

This command and function uninstalls a previously installed [Corel Add-on](#) (.CAO) from VENTURA.

Syntax	Description
<b>ReturnValue</b>	Specifies the numeric variable that is assigned a value corresponding to whether the specified Add-on was uninstalled: TRUE (-1) if not uninstalled; otherwise FALSE (0).
<b>.CAOName</b>	Specifies the name of the Add-on to uninstall.

### Note

**I** This command corresponds to the Corel Add-ons dialog box in Corel VENTURA. Click Tools, Add-ons, select an Add-on, Uninstall.

**I** Uninstalling a Corel Add-on does not remove it from your hard drive or system.

### Example

```
.CAOUninstall "CarSpec"
```

The above example uninstalls the Corel Add-on CARSPEC. This line can also be specified as follows:

```
ReturnValue = .CAOUninstall ("CarSpec")
```

---

{button ,AL("VENTURA\_CustomGetMenuAt\_Menu;vent\_customize;;;";0,"Defaultoverview",)} [Related Topics](#)



## .CustomAddButton (VENTURA)

**.CustomAddButton .Toolbar=string, .Group=string, .Command=string, .ButtonFace=long**

This command adds a button to a Corel VENTURA toolbar.

Syntax	Description
<b>.Toolbar</b>	Specifies the name of the toolbar to add the button to. This name corresponds to a name in VENTURA's Toolbars dialog box (Click View, Toolbars). A new toolbar is created if the specified toolbar does not exist.
<b>.Group</b>	Specifies the command category in which the command is grouped. The command categories correspond to the Command Categories list box in VENTURA's Customize dialog box (Click Tools, Customize, Toolbars tab).
<b>.Command</b>	<p>Specifies the name of the command. This name corresponds to a command name listed in a Command categories folder. For a listing of the command names in a Command category, click Tools, Customize, Toolbars tab, and select a Command categories folder.</p> <p>The <b>.CustomAddButton</b> command can only be used to add dynamic commands; that is, commands that are unique to the user's system. For example, the Command Category groups Font, Ventura Scripts, Document Scripts, Extended Characters, DBPub Recipes, Paragraph tags, and Add-ons all contain commands that are unique to a user's system.</p>
<b>.ButtonFace</b>	<p>Specifies the custom button to use for the command. Buttons are specified using a button index number. The first custom button in the Customize dialog box (Click Tools, Customize, Toolbars tab) corresponds to 0, the second custom button to 1, and so on. Custom buttons are numbered from left to right.</p> <p>If a button index number is not specified, a text button is used for the command.</p>

### Note



This command cannot be recorded.



The first three parameters used in this command are case-sensitive. See the VENTURA's Customize dialog box (Click Tools, Customize) for the correct spellings of parameters.



Including this command in a Corel Add-on can automatically assign an Add-on function to a toolbar button.

### Example

```
.CustomAddButton "My Fonts", "Font", "Courier"
```

The above example adds a button to a custom toolbar named My Fonts. This button applies the Courier font to selected text.

```
.CustomAddButton "Standard", "Ventura Scripts", "Dbpub.csc"
```

The above example adds a button to the Standard toolbar that starts the Dbpub script.

```
.CustomAddButton "Standard", "Add-ons", "(Run)Internet", 2
```

The above example adds a button to the Standard toolbar that executes the **(Run)Internet** Corel Add-on function. The third custom button (telephone bitmap) is used as the button.

---

{button ,AL('VENTURA\_CustomAddButton\_Menu;vent\_customize;;;','0,"Defaultoverview",)} [Related Topics](#)

## .CustomAddMenu (VENTURA)

**.CustomAddMenu** *.Menu=string, .Group=string, .Command=string, .MenuItem=string, .Position=long*

This command adds a menu command to a Corel VENTURA menu.

Syntax	Description
<b>.Menu</b>	Specifies the name of the menu to add the command to. This name corresponds to a command category in the Command list box in VENTURA's Customize dialog box (Click Tools, Customize, Menu tab). A new menu is created if the specified menu does not exist.
<b>.Group</b>	Specifies the command category the command is grouped in. The command categories correspond to the Command list box in VENTURA's Customize dialog box (Click Tools, Customize, Menu tab).
<b>.Command</b>	<p>Specifies the name of the command. This name corresponds to a command name listed in a Command folder. For a listing of the command names in a command folder, click Tools, Customize, Menu tab, and select a command folder.</p> <p>The <b>.CustomAddMenu</b> command can only be used to add dynamic commands; that is, commands that are unique to the user's system. For example, the command groups Font, Ventura Scripts, Document Scripts, Extended Characters, DBPub Recipes, Paragraph tags, and Add-ons all contain commands that are unique to a user's system.</p>
<b>.MenuItem</b>	Specifies the text to display for the menu item. If not specified, the name specified in <b>.Command</b> is used.
<b>.Position</b>	Specifies the position of the menu item in the menu. The first position is 0, the second position is 1, and so on. If not specified, the menu item is placed in the last position on the menu.

### Note



This command cannot be recorded.



The first three parameters used in this command are case-sensitive. See the VENTURA's Customize dialog box (Click Tools, Customize) for the correct spellings of parameters.



Including this command in a Corel Add-on can automatically assign an Add-on function to a menu command when the Add-on is installed.



Including this command in a Corel Add-on can automatically assign an Add-on function to a menu command.

### Example

```
.CustomAddMenu "My Fonts", "Font", "Courier"
```

The above example adds a command to a custom menu named My Fonts. This menu item applies the Courier font to selected text.

```
.CustomAddMenu "Tools", "Ventura Scripts", "Dbpub"
```

The above example adds a command to the Tools menu that starts the Dbpub script.

```
.CustomAddMenu "Tools", "Add-ons", "(Run)Internet", "Internet...", 1
```

The above example adds a command to the Tools menu that executes the **(Run)Internet** Corel Add-on function. The menu item is displayed as **Internet...** and is positioned as the second item on the Tools menu.

---

{button ,AL('VENTURA\_CustomAddButton\_Menu;vent\_customize;;;','0,"Defaultoverview",)} [Related Topics](#)

## .CustomGetCommandAt (VENTURA)

**.CustomGetCommandAt** *.MenuIndex=long, .CommandIndex=long, .CommandName=string, .Accelerator=string*

This function assigns the command names and shortcut keys used in a specified main menu in VENTURA. For example, the default main menu in VENTURA includes File, Edit, View,... .

Syntax	Description
<b>.MenuIndex</b>	Specifies a main menu's index number. A menu's index number corresponds to its position. The first menu corresponds to 1, the second corresponds to 2, and so on.
<b>.CommandIndex</b>	Specifies a command's index number. A command's index number corresponds to its position in the menu. The first command corresponds to 1, the second corresponds to 2, and so on. Flyout menu commands are not included in this index.
<b>.CommandName</b>	Specifies the string variable that is assigned the specified command's name.
<b>.Accelerator</b>	Specifies the string variable that is assigned the specified command's shortcut key, if it exists. The shortcut formatting is similar to the following shortcut keys: Ctrl+C Alt+Shift+C

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.



The CUSTOMKB.CSC script included with VENTURA creates a listing of accelerator keys used in VENTURA.

### Example

```
.CustomGetCommandAt 2, 1, Cname$, SKey$
```

In the above example, the first command name in the second VENTURA menu is assigned to the **Cname** variable. Its shortcut key is assigned to the string variable **SKey**.

---

{button ,AL("VENTURA\_CustomGetCommandAt\_Menu;;;;",0,"Defaultoverview"),} [Related Topics](#)

## .CustomGetMenuAt (VENTURA)

**.CustomGetMenuAt** *.MenuIndex=long*, *.MenuName=string*, *.NumOfCommands=long*

This function assigns the menu name and the number of commands of a specified main menu in VENTURA. For example, the default main menu in VENTURA includes File, Edit, View,... .

Syntax	Description
<b>.MenuIndex</b>	Specifies a main menu's index number. A menu's index number corresponds to its position. The first menu corresponds to 1, the second corresponds to 2, and so on.
<b>.MenuName</b>	Specifies the string variable that is assigned the specified main menu's name.
<b>.NumOfCommands</b>	Specifies the numeric variable that is assigned the number of commands on the specified main menu. Flyout menu commands are not included in this number.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
NumberMenu& = .CustomGetMenuCount ()  
.CustomGetMenuAt NumberMenu&, MenuName$, NumberCommands&
```

The first line assigns the number of main menus in VENTURA to the **NumberMenu** variable. The second line assigns the name of the last main menu in VENTURA and the number of commands on this menu to the variables **MenuName** and **NumberCommands**.

---

{button ,AL("VENTURA\_CustomGetMenuAt\_Menu;vent\_customize;;;",0,"Defaultoverview",)} Related Topics


# .CustomGetMenuCount (VENTURA)

ReturnValue& = .CustomGetMenuCount ( )

This function returns the number of main menus in the current VENTURA setup. For example, the default main menu in VENTURA includes File, Edit, View,... .

Syntax	Description
ReturnValue&	Specifies the numeric variable that is assigned the number of main menus in VENTURA.

### Note

 This command cannot be recorded.

### Example

```
NumberMenu& = .CustomGetMenuCount ( )  
.CustomGetMenuAt NumberMenu&, MenuName, NumberCommands
```

The first line assigns the number of main menus in VENTURA to the **NumberMenu** variable. The second line assigns the name of the last main menu in VENTURA and the number of commands on this menu to the variables **MenuName** and **NumberCommands**.

---

{button ,AL('VENTURA\_CustomGetMenuCount\_Menu;vent\_customize;;;',0,"Defaultoverview",,)} [Related Topics](#)

## **.CustomShowToolbar (VENTURA)**

**.CustomShowToolbar** **.Toolbar=string**, **.Visible=Boolean**

This command displays and hides toolbars in VENTURA.

Syntax	Description
<b>.Toolbar</b>	Specifies the name of the toolbar to display or hide. This name corresponds to a name in the Toolbars dialog box (Click View, Toolbars).
<b>.Visible</b>	Specifies whether to display or hide the toolbar. Set to TRUE (-1) to display (default if omitted); set to FALSE (0) to hide.

### **Note**



This command corresponds to the Toolbars dialog box. Click View, Toolbars. This command cannot be recorded.

### **Example**

```
.CustomShowToolbars "Recording And Script", TRUE
```

The above example displays the Recording And Script VENTURA toolbar.

---

`{button ,AL('VENTURA_CustomShowToolbar_Menu;vent_customize;;;','0',"Defaultoverview"),}` [Related Topics](#)

## **.EditAlign (VENTURA)**

**.EditAlign .Type=*long***

This command changes the alignment of the currently selected graphic objects.

Syntax	Description
<b>.Type</b>	Specifies the type of the alignment.
0	Left
1	Right
2	Top
3	Bottom
4	Horizontal
5	Vertical

### **Example**

```
.EditAlign 0
```

This example aligns the currently selected objects to the left.

---

{button ,AL(^VENTURA\_EditAlign\_Menu;;;;;'0,"Defaultoverview",)} Related Topics

## **.EditCanRedo (VENTURA)**

**ReturnValue** = .EditCanRedo ( )

This function returns a value corresponding to whether the last operation can be redone.

Syntax	Description
<b>ReturnValue</b>	Specifies the numeric variable that is assigned a value corresponding to whether the last operation can be redone: TRUE (-1) if it can be redone; otherwise FALSE (0).

### **Note**



This command cannot be recorded.



See the [.EditRedo](#) command for more information.

### **Example**

```
RV_Redo = .EditCanRedo
```

In the above example, the return value is assigned to the **RV\_Redo** variable.

---

{button ,AL('VENTURA\_EditCanRedo\_Menu;vent\_edit;;;','0,"Defaultoverview",')} [Related Topics](#)



## **.EditCanUndo (VENTURA)**

**ReturnValue** = .EditCanUndo ( )

This function returns a value corresponding to whether the last operation can be undone.

Syntax	Description
<b>ReturnValue</b>	Specifies the numeric variable that is assigned a value corresponding to whether the last operation can be undone: TRUE (-1) if it can be undone; otherwise FALSE (0).

### **Note**



This command cannot be recorded.



See the [\*\*.EditUndo\*\*](#) command for more information.

### **Example**

```
RV_Undo = .EditCanUndo
```

In the above example, the return value is assigned to the **RV\_Undo** variable.

---

{button ,AL('VENTURA\_EditCanUndo\_Menu;vent\_edit;;;','0',"Defaultoverview"),} [Related Topics](#)

## **.EditCenter (VENTURA)**

**.EditCenter** *.Vertical=Boolean*

This command centers the selected graphic objects.

Syntax	Description
.Vertical	Set to TRUE (-1) to center vertically. Set to FALSE (0) to center horizontally. The default is FALSE.

### **Example**

```
.EditCenter TRUE
```

This example centers the selected graphic objects vertically.

---


{button ,AL('VENTURA\_EditCenter\_Menu;;;;','0',"Defaultoverview"),} [Related Topics](#)


## **.EditConvertShapeToFrame (VENTURA)**

### **.EditConvertShapeToFrame**

This command converts closed graphic objects (drawn with the Graphics tool) into a frame.

#### **Note**

 This command corresponds to the Convert to Frame command which can be accessed by a right-click when a graphic is selected in Corel VENTURA.

 You can also convert graphic objects that are created using the Convert Picture to Shapes command which can be accessed by a right-click when a graphic is selected. See the [.EditConvertVectorToShape](#) command for more information.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 0
.FileImportPicture "c:\MyPics\tree.cdr", 104, FALSE
.EditConvertVectorToShape
.EditConvertShapeToFrame
```

The above example imports a CorelDRAW file named TREE.CDR into a frame. The TREE picture is converted into a graphic and then the graphic is converted into a frame.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_EditCanUndo\_Menu;vent\_edit;;;','0,"Defaultoverview"),} [Related Topics](#)

## **.EditConvertToCurves (VENTURA)**

### **.EditConvertToCurves**

Converts the selected frame or graphic object into a series of curves and/or lines that can be edited using the Node tool. Once an object is converted to curves, it is not possible to return it to its original object type, except by using the Undo command in the Edit menu (see the [.EditUndo](#) command).

#### **Note**



This command corresponds to clicking Format, Arrange, Convert to Curves in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.EditConvertToCurves
```

The above example draws a rectangle and converts it into a series of curved lines.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_EditConvertToCurves\_Menu;vent\_edit;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.EditConvertToPresetShape (VENTURA)**

**.EditConvertToPresetShape** .ShapeType=*long*

This command converts the shape of the selected graphic object or frame to the given preset style.

Syntax	Description
.ShapeType	Specifies the type of the preset shape.
0	Rectangle (default if omitted)
1	Inverted rectangle
2	Rounded rectangle
3	Left trapezoid
4	Right trapezoid
5	L90
6	L270
7	Heart
8	Octagon
9	Tear drop
10	Star
11	Bubble
12	Ellipse
13	Diamond

### **Example**

`.EditConvertToPresetShape 7`

This example converts the shape of the selected graphic object or frame to a heart.

---

{button ,AL('VENTURA\_EditConvertToPresetShape\_Menu;;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.EditConvertVectorToShape (VENTURA)**

### **.EditConvertVectorToShape**

This command converts imported pictures in vector format into objects that you can fill, outline and modify with the Node Edit tool



#### **Note**



This command corresponds to the Convert Picture to Shapes command which can be accessed by a right-click when a vector picture is selected in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 0
.FileImportPicture "c:\MyPics\tree.cdr", 104, FALSE
.EditConvertVectorToShape
.EditConvertShapeToFrame
```

The above example imports a CorelDRAW file named TREE.CDR into a frame. The TREE picture is converted into a graphic and then the graphic is converted into a frame.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_EditCanUndo\_Menu;vent\_edit;;;','0,"Defaultoverview"),} **Related Topics**

## **.EditCopy (VENTURA)**

### **.EditCopy**

This command copies the current selected text, frame or graphic objects from the active document and places it on the Clipboard.

#### **Note**



This command corresponds to the Copy command in the Edit menu in Corel VENTURA. Click Edit, Copy.

#### **Example**

```
.TextWordRight 3, -1  
.EditCopy  
.TextParaDown 1,  
.EditPaste
```

The above example selects the next three words, including the current word, and copies them to the Clipboard. The words are then pasted at the beginning of the next paragraph.

---

{button ,AL("VENTURA\_EditCopy\_Menu;vent\_edit;;;",0,"Defaultoverview",)} Related Topics

## **.EditCut (VENTURA)**

### **.EditCut**

This command removes the current selected text, frame or graphic objects from a document and places it on the Clipboard.

#### **Note**



This command corresponds to the Cut command in the Edit menu in Corel VENTURA. Click Edit, Cut.



Use the **.EditDelete** command to remove a selection and not have it placed it on the Clipboard.

#### **Example**

```
.TextCharRight 1, -1  
.EditCut  
.TextCharRight 1  
.EditPaste
```

This example reverses two characters by selecting and cutting the first one to the Clipboard and pasting it to the right of the next character.

---

{button ,AL("VENTURA\_EditCut\_Menu;vent\_edit;;;",0,"Defaultoverview"),} [Related Topics](#)



## **.EditDelete (VENTURA)**

### **.EditDelete**

This command removes the current selected text, frame or graphic object from a document. The selection is not placed on the Clipboard. To place the removed selection on the Clipboard, use the **.EditCut** command instead of the **.EditDelete** command.

#### **Note**



This command corresponds to the Delete command in the Edit menu in Corel VENTURA. Click Edit, Delete.

#### **Example**

```
.TextCharRight 1, -1  
.EditDelete
```

This example selects and deletes the character to the right of the insertion point.

---

{button ,AL('VENTURA\_EditDelete\_Menu;vent\_edit;;;','0',"Defaultoverview"),} [Related Topics](#)

## **.EditDuplicate (VENTURA)**

### **.EditDuplicate**

Duplicates the selected frame or graphic object. Use the Tools, Options command (Tolerances tab) to set the amount the duplicate is offset from the original.

#### **Note**



This command corresponds to the Duplicate command in the Edit menu in Corel VENTURA. Click Edit, Duplicate.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.EditDuplicate
.EditOrderBackOne
```

The above example draws an ellipse and duplicates it, leaving two ellipses. The duplicate ellipse is then positioned behind the original ellipse in the object stacking order.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Edit\_Menu;vent\_edit;;;','0,"Defaultoverview"),} **Related Topics**

## **.EditEditSpecialItem (VENTURA)**

**.EditEditSpecialItem**

### **Note**



This command was introduced in Corel VENTURA 8.

### **Example**

The

## **.EditGroup (VENTURA)**

### **.EditGroup**

This command groups all selected frames of graphic objects together so they can be selected and manipulated as a single object.

#### **Note**



This command corresponds to clicking Format, Arrange, Group in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4*M_INCH, 4*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.SelectObjectAt 2*M_INCH, 2*M_INCH, TRUE
.EditGroup
.RotateObject 450
.EditUngroup
```

The above example draws a rectangle and an ellipse. While the ellipse is already selected, the rectangle is also selected. Both objects are then grouped as a single object and rotated 45 degrees. The objects are then ungrouped.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Arrange\_Menu;vent\_edit;;;",0,"Defaultoverview",)} **Related Topics**

## **.EditMakeSameSize (VENTURA)**

**.EditMakeSameSize** .Width=*Boolean*, .Height=*Boolean*

This command makes the height, width, or both of selected objects identical. The height and width of the last selected object is used to set size.

Syntax	Description
.Width	Set to TRUE (-1) to make the widths the same. Otherwise, set to FALSE (0). The default is TRUE.
.Height	Set to TRUE (-1) to make the heights the same. Otherwise, set to FALSE (0). The default is TRUE.

### **Note**



This command cannot be recorded.

### **Example**

```
.EditMakeSameSize TRUE, TRUE
```

This example makes the height and width of all selected objects the same.

---

`{button ,AL("VENTURA_EditMakeSameSize_Menu;;;;",0,"Defaultoverview",)}` [Related Topics](#)

## **.EditOrderBackOne (VENTURA)**

### **.EditOrderBackOne**

This command rearranges the object stacking order by moving the selected frame or graphic object down one position (to the back of the screen).

#### **Note**



This command corresponds to clicking Format, Arrange, Back One in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileNew 1
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.EditDuplicate
.EditDuplicate
.EditOrderBackOne
```

The above example draws an ellipse and duplicates it twice, leaving the second duplicate on top of the other two ellipses in the stacking order. The second duplicate ellipse, still selected, is sent back one position in the stacking order behind the first duplicate ellipse.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_EditOrderBackOne\_Menu;vent\_edit;;;',0,"Defaultoverview",)} Related Topics

## **.EditOrderForwardOne (VENTURA)**

### **.EditOrderForwardOne**

This command rearranges the object stacking order by moving the selected frame or graphic object up one position (to the front of the screen).

#### **Note**



This command corresponds to clicking Format, Arrange, Forward One in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.DrawEllipse 1.75*M_INCH, 1.75*M_INCH, 1*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.EditOrderForwardOne
```

The above example draws a rectangle and two ellipses which are placed on top of the rectangle. The rectangle is then selected using the **.SelectObjectAt** command and is ordered one position forward in the stacking order so that it is between the two ellipses.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_EditOrderForwardOne\_Menu;vent\_edit;;;','0,"Defaultoverview",)} **Related Topics**

## **.EditOrderReverseOrder (VENTURA)**

### **.EditOrderReverseOrder**

This command reverses the stacking order of two or more selected frames or graphic objects.

#### **Note**



This command corresponds to clicking Format, Arrange, Reverse Order in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileNew 1
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.EditDuplicate
.EditDuplicate
.EditSelectAll
.EditOrderReverseOrder
```

The above example draws an ellipse and duplicates it twice, leaving the second duplicate on top of the other two ellipses in the stacking order. All three ellipses are selected and their stacking order is inverted so that the original ellipse is on top.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_EditOrderReverseOrder\_Menu;vent\_edit;;;','0,"Defaultoverview",)} **Related Topics**



## **.EditOrderToBack (VENTURA)**

### **.EditOrderToBack**

This command places the selected frame or graphic object in back of other objects.

#### **Note**



This command corresponds to clicking Format, Arrange, To Back in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileNew 1
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.EditDuplicate
.EditDuplicate
.EditOrderToBack
```

The above example draws an ellipse and duplicates it twice, leaving the second duplicate on top of the other two ellipses in the stacking order. The second duplicate ellipse, still selected, is sent to the back of the stacking order.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_EditOrderToBack\_Menu;vent\_edit;;;',0,"Defaultoverview",)} **Related Topics**

## **.EditOrderToFront (VENTURA)**

### **.EditOrderToFront**

This command places the selected frame or graphic object in front of other objects.

#### **Note**



This command corresponds to clicking Format, Arrange, To Front in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.EditOrderToFront
```

The above example draws a rectangle and an ellipse which is placed on top of the rectangle. The rectangle is then selected using the **.SelectObjectAt** command and is ordered to the top of the stacking order.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_EditOrderToFront\_Menu;vent\_edit;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.EditPaste (VENTURA)**

### **.EditPaste**

This command places a copy of the Clipboard contents into your document at the insertion point. This command can also be used to paste frames and graphics depending on the status of the pointer.

#### **Note**



This command corresponds to the Paste command in the Edit menu in Corel VENTURA. Click Edit, Paste.



The original cut or copied object remains on the Clipboard until you copy or cut another object, or end the current Windows session.



You can also use the **.EditPasteLink** and **.EditPasteSpecial** commands to paste the Clipboard contents.

#### **Example**

```
.TextCharRight 1, -1  
.EditCut  
.TextCharRight 1  
.EditPaste
```

This example reverses two characters by selecting and cutting the first one to the Clipboard and pasting it to the right of the next character.

---

**{button ,AL("VENTURA\_EditPaste\_Menu;vent\_edit;;;','0,"Defaultoverview",,)} Related Topics**

## **.EditRedo (VENTURA)**

### **.EditRedo**

This command reverses the **.EditUndo** command (or the Edit, Undo command) and reapplies the last operation.

#### **Note**



This command corresponds to the Redo command in the Edit menu in Corel VENTURA. Click Edit, Redo.

#### **Example**

`.EditRedo`

---

`{button ,AL("VENTURA_EditRedo_Menu;vent_edit;;;",0,"Defaultoverview",)} Related Topics`

## **.EditRepeat (VENTURA)**

### **.EditRepeat**

This command repeats the last repeatable change that was made to the document. Most editing and formatting actions can be repeated.

#### **Example**

```
.TextWordRight 1, TRUE  
.FormatSelectedText .Weight = 700  
.TextCharRight 1, FALSE  
.TextWordRight 1, TRUE  
.EditRepeat
```

The above example selects a word and applies bold formatting to it. The word to the right of the selected word is then selected and the same formatting is applied to it.

---

{button ,AL("VENTURA\_EditRepeat\_Menu;vent\_edit;;;",0,"Defaultoverview",)} [Related Topics](#)

## **.EditSelectAll (VENTURA)**

### **.EditSelectAll**

Depending on the current Corel VENTURA mode or what is selected, this command performs different selection functions.



In text mode, if the insertion point is on the base page, the command selects all the text on all the base pages.



If a frame or a graphic is selected on the active page, the command selects all other graphic and frames on the active page.



If a table cell is selected, this command selects the rest of the cells in the table.

#### **Note**



For other selection commands, see the [.FrameSelectAll](#) and [.TableSelectTable](#) commands.



This command corresponds to the Select All command in Corel VENTURA's Edit menu. Click Edit, Select All.

#### **Example**

`.EditSelectAll`

---

`{button ,AL("VENTURA_EditRepeat_Menu;vent_edit;;;','0,"Defaultoverview",)} Related Topics`

## **.EditSelectAllLinked (VENTURA)**

### **.EditSelectAllLinked**

This command selects all frames that are linked to the currently selected frame (i.e., the frames with the same text file).

### **Example**

`.EditSelectAllLinked`

This example selects all frames that are linked to the currently selected frame.

---

`{button ,AL('VENTURA_EditSelectAllLinked_Menu;;;;',0,"Defaultoverview"),}` [Related Topics](#)

## **.EditSpace (VENTURA)**

**.EditSpace** *.Across=Boolean*

This command equalizes the horizontal or vertical spacing between three or more selected frames or graphic objects.

Syntax	Description
.Across	Set to TRUE (-1) to add space across. Set to FALSE (0) to add space down. The default is FALSE.

### **Example**

```
.EditSpace TRUE
```

This example equalizes the horizontal spacing of the selected objects.

---

{button ,AL('VENTURA\_EditSpace\_Menu;;;;','0',"Defaultoverview"),} Related Topics



## **.EditUndo (VENTURA)**

### **.EditUndo**

This command cancels the last undoable change that was made to the document and returns it to its former state. Most editing and formatting actions can be undone.

#### **Note**



This command corresponds to the Undo command in the Edit menu in Corel VENTURA. Click Edit, Undo.

#### **Example**

`.EditUndo`

The above example cancels the last change that was made to the document.

---

`{button ,AL("VENTURA_EditUndo_Menu;vent_edit;;;','0,"Defaultoverview"),}` [Related Topics](#)

## **.EditUngroup (VENTURA)**

### **.EditUngroup**

This command breaks up the selected group into its individual objects. If you have more than one sub-level of grouping, **.EditUngroup** breaks up one level of grouping at a time.

#### **Note**



This command corresponds to clicking Format, Arrange, Ungroup in Corel VENTURA.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4*M_INCH, 4*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.SelectObjectAt 2*M_INCH, 2*M_INCH, TRUE
.EditGroup
.RotateObject 450
.EditUngroup
```

The above example draws a rectangle and an ellipse. While the ellipse is already selected, the rectangle is also selected. Both objects are then grouped as a single object and rotated 45 degrees. The objects are then ungrouped.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Arrange\_Menu;vent\_edit;;;",0,"Defaultoverview",)} [Related Topics](#)

## .FileClose (VENTURA)

**.FileClose** .SaveOption=*long*, .AllDocuments=*Boolean*

This command closes COREL VENTURA publications.

Syntax	Description
.SaveOption	Specifies saving options before closing publications: 0 Opens dialog box asking using whether to save changes (default if parameter is omitted). 1 Save changes made to the publication(s). If the publication(s) has not been previously saved, the Save as dialog box opens. 2 Close the publication without saving.
.AllDocuments	Specifies whether to close all open COREL VENTURA publications or the active publication. Set to TRUE (-1) to close all publications. Set to FALSE (0) to close the active publication.

### Note



This command corresponds to the Close command in the File menu in Corel VENTURA. Click File, Close.

### Example

```
.FileClose 2, -1
```

The above example closes all open COREL VENTURA publications without saving.

---

{button ,AL('vent\_file;;;','0',"Defaultoverview",)} Related Topics

## **.FileExit (VENTURA)**

**.FileExit** *.SaveOption=long*

This command ends a Corel VENTURA session.

Syntax	Description
.SaveOption	Specifies saving options for open publications before exiting COREL VENTURA: 0 Opens dialog box asking using whether to save changes (default if parameter is omitted). 1 Save changes made to the publication(s). If the publication(s) has not been previously saved, the Save as dialog box opens. 2 Close the publication without saving.

### **Note**



This command cannot be recorded.



This command corresponds to the Exit command in the File menu in Corel VENTURA. Click File, Exit.

### **Example**

`.FileExit 1`

The above example ends a Corel VENTURA session and saves changes to the open publications.

---

`{button ,AL('vent_file;;;;',0,"Defaultoverview",)} Related Topics`

## .FileExport (VENTURA)

**.FileExport** **.FileName**=*string*, **.FilterType**=*long*, **.Markup**=*Boolean*

This command exports the active text file in the current chapter in industry-standard ASCII, ANSI and RTF formats. This command was modified in Corel VENTURA 8.

Syntax	Description
<b>.FileName</b>	Specifies the name of the text file to export. If the path is not specified, the active folder is used to save the file.
<b>.FilterType</b>	Specifies the filter to be used for export: 11 Rich Text Format (.RTF) 51 ANSI text (.TXT) 52 ASCII text (.TXT) 53 ASCII 8-bit text (.TXT)
<b>.Markup</b>	Specifies whether to include Corel VENTURA markup text codes in the exported text. Set to TRUE (-1) to enable this option; otherwise FALSE(0). If not specified, set to FALSE. This parameter is ignored when exporting to RTF format. This parameter is not used when the filter type is set to RTF (11).

### Note



This command corresponds to the Export command in the File menu in Corel VENTURA. Click File, Export.

### Example

```
.FileExport "C:\VENTURA\EXPORTS\NEWFILE.RTF" , 11
```

The above example exports the active text file to NEWFILE.RTF as a rich text format.


---

{button ,AL('vent\_file;;;','0',"Defaultoverview"),} Related Topics


## .FileImportText (VENTURA)


**.FileImportText** **.FileName=string**, **.FilterType=long**, **.ReplaceOption=long**, **.InsertAtCursor=Boolean**, **.AddToList=Boolean**

This command imports text in many word processor file formats and from programs that save text in industry-standard ASCII, ANSI and RTF (Rich Text Format) format. This command was modified in Corel VENTURA 8.


Syntax	Description
<b>.FileName</b>	Specifies the name of the document to open. If path information is not included, the active folder is used.
<b>.FilterType</b>	In Corel VENTURA 7, this parameter specified a value corresponding to a filter type. In Corel VENTURA 8, this parameter should be set to 0  Corel VENTURA determines and uses the appropriate filter. If omitted, it is set to 0.
<b>.ReplaceOption</b>	Specifies an option if a file with the same name as that specified in <b>.FileName</b> exists in the chapter: 0 Auto rename the file being imported. For example FILE.TXT may become FILE1.TXT. If this parameter is not specified, this is the default setting. 1 Replace existing file 2 Append to existing file
<b>.InsertAtCursor</b>	Specifies whether to place the selected text file at the insertion point. The file you are importing merges with the text file containing the insertion point. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted.
<b>.AddToList</b>	Specifies whether to put the specified file in the File List only rather than in the selected frame. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted. If set to FALSE, the imported file replaces the one in the selected frame.

### Note

 If a frame is selected before this command is issued, the imported text is placed in the selected frame. If a frame isn't selected before this command is issued, the imported text can be attached to a frame with the **.FormatAttachTextFile** command.

 The **AddToList** and **InsertAtCursor** parameters are mutually exclusive. The **AddToList** parameter is ignored if the **InsertAtCursor** parameter is used.

 This command corresponds to the Import Text command in the File menu in Corel VENTURA. Click File, Import Text.

 Pictures can be imported with the **.FileImportPicture** command.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileImportText "C:\MYDOCS\REPORT.WP6", 0, 1
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatAttachTextFile "REPORT.WP6"
```

The above example imports the Corel WordPerfect document REPORT.WP6 into the active chapter. The **.FormatAttachTextFile** command is used to select an already existing frame, and the imported WordPerfect document is attached to it.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('vent\_file;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FileNew (VENTURA)

**.FileNew** .TemplateName=*string*

This command creates a new Corel VENTURA publication. The new publication is active.

Syntax	Description
.TemplateName	Specifies the path and file name of the template to use. If this parameter is not specified, the DEFAULT.VP template is used.

### Note



This command corresponds to the New command in the File menu in Corel VENTURA. Click File, New.

### Example

`.FileNew`

The above example creates a new COREL VENTURA publication using the DEFAULT template.

`.FileNew "C:\Corel\Graphics8\Ventura\Samples\Books\book.vp"`

The above example creates a new COREL VENTURA publication using the BOOK.VP publication as a template.

---

`{button ,AL('vent_file;;;','0',"Defaultoverview"),}` [Related Topics](#)

## .FileOpen (VENTURA)

**.FileOpen** **.FileName=string**, **.NewFileName=string**, **.UseAllTags=Boolean**, **.RenameStyle=long**, **.UseUnderscore=Boolean**, **.ShowPanoseDialog=Boolean**, **.MaxPictureSize=long**

This command loads a previously saved publication into Corel VENTURA. This command can open Corel VENTURA 7 and 8 publications, pre-VENTURA 7 publications and chapters, and a variety of word processor files including Corel WordPerfect files. For a list of the file types this command can open, in Corel VENTURA click File, Open, and then click the Files of type list box.

Syntax	Description
<b>.FileName</b>	Specifies the name of the publication to open. If path information is not included, the active folder is used. If the publication specified is already open, it becomes the active publication.
<b>.NewFileName</b>	This parameter is only valid when opening a Ventura 3, 4, or 5 publication. Pre-VENTURA 7 publications must be converted and saved to the Corel VENTURA 8 format before being used. This parameter specifies the filename of the converted file. If path information is not included, the active folder is used.
<b>.UseAllTags</b>	In VENTURA 3, 4, or 5 publications that use more one style sheet, specifies whether to use all tags or only one set of tags. Set to TRUE (-1) to use all tags. Set to FALSE (0) to use one set of tags. This parameter is only valid when opening a Ventura 3, 4, or 5 publication.
<b>.RenameStyle</b>	If the <b>.UseAllTags</b> parameter is set to TRUE, specifies the duplicate tag renaming style: 0     Style name with the suffix (default if omitted) 1     Style number with the suffix 2     Style name with the prefix 3     Style number with the prefix
<b>.UseUnderscore</b>	If the <b>.UseAllTags</b> parameter is set to TRUE, specifies whether to use the underscore character (_) as a separator for the prefix/suffix in the duplicate tag names. Set to TRUE (-1) to use the underscore character; otherwise, set to FALSE.
<b>.ShowPanoseDialog</b>	If the user's system does not have a font the publication to be opened uses, this parameter specifies whether to open the PANOSE Font Matching Results dialog box. Set TRUE (-1) to open the dialog box; otherwise, set to FALSE (0) which is the default if omitted. Setting this parameter to FALSE (0) corresponds to clicking OK on the PANOSE Font Matching Results dialog box.
<b>.MaxPictureSize</b>	This parameter is only valid when opening a Ventura 3, 4, or 5 publication. Pre-VENTURA 7 publications must be converted and saved to the VENTURA 7 format before being used. This parameter specifies the maximum size of an embedded picture in kilobytes (KB).

### Note



This command corresponds to the Open command in the File menu in Corel VENTURA. Click File, Open.



The publication opened with the **.FileOpen** command becomes the active publication.

### Example

```
.FileOpen "C:\VENTURA\MYFILES\TEST1.VP"
```

The above example opens a COREL VENTURA publication named TEST1.VP

```
.FileOpen "C:\Corel\WP\INTERNAL.WP6"
```

The above example opens a Corel WordPerfect 6 file named INTERNAL.WP6.

```
.FileOpen "C:\VENTURA5\ABC_CO.PUB", "C:\VENTURA7\ABC_CO.VP". -1, 2, -1
```

The above example opens a VENTURA 5 publication named ABC\_CO.PUB that uses multiple style sheets. The publication is converted to a VENTURA 8 format, and all the tags from the style sheets are kept, and duplicate tags are renamed using the style name as the prefix with the underscore character.

```
.FileOpen "C:\VENTURA\MYFILES\TEST1.VP", , , , TRUE
```

The above example opens a COREL VENTURA publication named TEST1.VP and opens the PANOSE Font Matching Results dialog box if the user's system does a font the TEST1.VP publication uses installed.

---

{button ,AL('vent\_file;;;',0,"Defaultoverview"),} [Related Topics](#)



## .FilePrint (VENTURA)

**.FilePrint** *.Range=long, .FromNumber=long, .ToNumber=long, .Copies=long, .Collate=Boolean, .HiddenGraphics=Boolean, .PrinterName=string, .PrintToFile=string, .PrintChpToFile=Boolean, .PrintStyleName=string*

This command prints all or part of the active Corel VENTURA publication with specified options.

Syntax	Description
.Range	Specifies the print range: 0 Publication 1 Current page 2 Chapter range (see the <b>.FromNumber</b> and <b>.ToNumber</b> parameters) 3 Page range (see the <b>.FromNumber</b> and <b>.ToNumber</b> parameters) If this parameter is not specified, the current chapter is printed and all other parameters are ignored. See the <b>.FilePrintDefault</b> command for more information.
.FromNumber	Specifies the first page or chapter to print in a contiguous range of pages or chapters.
.ToNumber	Specifies the last page or chapter to print in a contiguous range of pages or chapters.
.Copies	Specifies the number of copies to print. Set to 1 if omitted.
.Collate	Specifies whether to collate multiple copies. Set to TRUE (-1) to enable collation; otherwise, set to FALSE (default if omitted).
.HiddenGraphics	Specifies whether to print hidden graphics. Set to TRUE (-1) to enable this option, otherwise, set to FALSE (0).
.PrinterName	Specifies the name of the printer to use. If not specified, the default system printer or the previously selected printer is used.
.PrintToFile	Specifies the name and path of the output file.
.PrintChpToFile	Specifies whether to print each chapter into a separate file. Set to TRUE (-1) to enable this option, otherwise, set to FALSE (0).
.PrintStyleName	Specifies the name of a print setting in the current COREL VENTURA system. A print setting holds predefined printing options. If this parameter is used, all other parameters are ignored.

### Notes



This command cannot be recorded.



This command corresponds to the Print command in the File menu in Corel VENTURA. Click File, Print.

### Example

```
.FilePrint 3, 120, 130, 4, 0, -1, "HP Laserjet"
```

The above example prints 4 copies of pages 120 to 130 in the current chapter. Hidden graphics are printed and the print job is not collated.

---

{button ,AL('vent\_file;;;','0',"Defaultoverview",)} **Related Topics**

## **.FilePrintDefault (VENTURA)**

### **.FilePrintDefault**

This command prints the active publication using the current print and printer settings.

#### **Note**



This command cannot be recorded.

#### **Example**

```
.FileprintDefault
```

---

{button ,AL("VENTURA\_FilePrintDefault\_Menu;vent\_file;;;','0,"Defaultoverview"),} Related Topics

## .FilePrintOptions (VENTURA)

**.FilePrintOptions** .PrintVectors=*Boolean*, .PrintBitmaps=*Boolean*, .PrintText=*Boolean*, .PrintTextInBlack=*Boolean*, .SeparateColor=*Boolean*, .PrintJobInfo=*Boolean*, .FountainSteps=*long*, .ColorType=*long*

This command specifies various printing options. If all parameters are omitted, this command opens the Print Setup dialog box.

Syntax	Description
.PrintVectors	Specifies whether to print vector pictures. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.PrintBitmaps	Specifies whether to print bitmap pictures. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.PrintText	Specifies whether to print text. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.PrintTextInBlack	Specifies whether to print text in black, if text is being printed. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.SeparateColor	Specifies whether to use color separations. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.PrintJobInfo	Specifies whether to print job information sheet after print job. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.FountainSteps	Specifies the number of fountain steps. If omitted, the default setting is used.
.ColorType	Specifies how to print color: 0 Prints using the full color capabilities of the selected printing device. 1 Print colors as black 2 Print color as gray scale.

### Note

**I** This command cannot be recorded.

### Example

```
.FilePrintOptions -1 ,-1, -1, , , , 130
```

The above example sets the printer options to print vector and bitmap pictures along with text. The fountain steps are set to 130.

---

{button ,AL('vent\_file;;;','0',"Defaultoverview"),} Related Topics

## .FilePrintOptionsLayout (VENTURA)

**.FilePrintOptionsLayout** .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long*, .CenterImage=*Boolean*, .FitToPage=*Boolean*, .LayoutStyle=*long*, .BleedLimit=*long*

This command sets printing layout settings.

Syntax	Description
.Left	Specifies the distance from the left edge of the printable page, in tenths of a micron.
.Top	Specifies the distance from the top edge of the printable page, in tenths of a micron.
.Width	Specifies the width of the printed publication, in tenths of a micron.
.Height	Specifies the height of the printed publication, in tenths of a micron. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.CenterImage	Specifies whether to center the publication on a printed page. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.FitToPage	Specifies whether to automatically scale the publication so that it fits on printable page. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.LayoutStyle	Specifies the preset page layouts: 0 Full Page (default if omitted) 1 Book 2 Booklet 3 Tent-card 4 Side-Fold card 5 Top-Fold card 6 Tri-Fold card 7 Manual
.BleedLimit	Specifies the bleed limit in tenths of a micron.

### Note

**I** This command cannot be recorded.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FilePrintOptionsLayout 1.5*M_INCH, 2*M_INCH, 6*M_INCH, 8*M_INCH, -1
```

The above example sets the left and top spaces to 1.5 and 2 inches, respectively. The printed publication is set to 6 by 8 inches and is centered on the page.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL(`vent\_file;;;',0,"Defaultoverview"),} [Related Topics](#)

## .FilePrintOptionsPosition (VENTURA)

**.FilePrintOptionsPosition** .Rows=*long*, .Columns=*long*, .LeftMargin=*long*, .RightMargin=*long*, .TopMargin=*long*, .BottomMargin=*long*, .AutoSpacing=*Boolean*, .HorzGutter=*long*, .VertGutter=*long*, .CloneFrame=*Boolean*, .DocPageSize=*Boolean*

This command specifies how to place several existing printable pages on a single printable page, or to place several copies of an existing printable page on a single page. This command also sets page margins.

Syntax	Description
.Rows	Specifies the number of rows of positioning frames to be placed on the printable page.
.Columns	Specifies the number of columns of positioning frames to be placed on the printable page.
.LeftMargin	Specifies the left margin, in the tenths of a micron.
.RightMargin	Specifies the right margin, in the tenths of a micron.
.TopMargin	Specifies the top margin, in the tenths of a micron.
.BottomMargin	Specifies the bottom margin, in the tenths of a micron.
.AutoSpacing	Specifies whether to use automatic gutter spacing. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.HorzGutter	Specifies the horizontal distance between each layout frame that is placed on the printable page. This parameter is ignored if <b>.AutoSpacing</b> is set to TRUE.
.VertGutter	Specifies the vertical distance between each layout frame that is placed on the printable page. This parameter is ignored if <b>.AutoSpacing</b> is set to TRUE.
.CloneFrame	Specifies whether to place the current working page in each frame of the printable page. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.DocPageSize	Specifies whether to maintain publication page size. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).

### Note

**I** This command cannot be recorded.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
FilePrintOptionsPosition 5, 4, 1.5*M_INCH, 2*M_INCH, 1*M_INCH, 1*M_INCH, -1
```

The above example prints 5 rows and 4 columns on a single page. The margins are 1.5, 2, 1, and 1 inch for left, right, top and bottom respectively. The gutters are automatically calculated.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('vent\_file;;;',0,"Defaultoverview",)} **Related Topics**

## .FilePrintOptionsSpecial (VENTURA)

**.FilePrintOptionsSpecial** .PrintFileInfo=*Boolean*, .PrintPageNum=*Boolean*, .CropMarks=*Boolean*, .RegMarks=*Boolean*, .CalibrationBar=*Boolean*, .Densitometer=*Boolean*, .PrintNegative=*Boolean*, .EmulsionDown=*Boolean*

This command sets printer mark and pre-press settings.

Syntax	Description
.PrintFileInfo	Specifies whether to print the name of file on each page. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.PrintPageNum	Specifies whether to print the page number on each page(this is different from the publication page numbers). Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.CropMarks	Specifies whether to print crop marks. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.RegMarks	Specifies whether to print registration marks. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.CalibrationBar	Specifies whether to print calibration bar. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.Densitometer	Specifies whether to print densitometer scale. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.PrintNegative	Specifies whether to print as a negative image. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).
.EmulsionDown	Specifies whether to enable emulsion mode. Set to TRUE (-1) to enable this printing option; otherwise, set to FALSE (0).

### Note



This command cannot be recorded.

### Example

```
.FilePrintOptionsSpecial -1, , -1
```

The above example enables printing file information and registration marks.

---

{button ,AL('vent\_file;;;',0,"Defaultoverview"),} Related Topics

## .FilePrintQuick (VENTURA)

**.FilePrintQuick** *.Range=long*, *.FromNumber=long*, *.ToNumber=long*, *.Copies=long*, *.Collate=Boolean*, *.PrinterName=string*, *.PrintToFile=string*

This command prints all or part of the active Corel VENTURA publication with specified options.

Syntax	Description
.Range	Specifies the print range: 0 Publication 1 Current page 2 Chapter range (see the <b>.FromNumber</b> and <b>.ToNumber</b> parameters) 3 Page range (see the <b>.FromNumber</b> and <b>.ToNumber</b> parameters) If this parameter is not specified, the current chapter is printed and all other parameters are ignored. See the <b>.FilePrintDefault</b> command for more information.
.FromNumber	Specifies the first page or chapter to print in a contiguous range of pages or chapters.
.ToNumber	Specifies the last page or chapter to print in a contiguous range of pages or chapters.
.Copies	Specifies the number of copies to print. Set to 1 if omitted.
.Collate	Specifies whether to collate multiple copies. Set to TRUE (-1) to enable collation; otherwise, set to FALSE (0) (default if omitted).
.PrinterName	Specifies the name of the printer to use. If not specified, the default system printer or the previously selected printer is used.
.PrintToFile	Specifies the name and path of the output file.

### Note

 This command cannot be recorded.

### Example

```
.FilePrint 3, 120, 130, 4, 0, -1, "HP Laserjet"
```

The above example prints 4 copies of pages 120 to 130 in the current chapter. The print job is not collated.

---

**{button ,AL('vent\_file;;;','0,"Defaultoverview",)} Related Topics**

## **.FilePublishAs (VENTURA)**

### **.FilePublishAs**

This command is only available in Corel VENTURA 7. Use one of the following commands to publish:

- [FilePublishAsAcrobat](#)
- [FilePublishAsBarista](#)
- [FilePublishAsCSS](#)
- [FilePublishAsHTML](#)

---

{button ,AL('vent\_file;;;','0',"Defaultoverview"),} [Related Topics](#)



## .FilePublishAsAcrobat (VENTURA)

**.FilePublishAsAcrobat** **.FileName=string**, **.PrinterName=string**

This command publishes the current publication to PostScript file which can then be distilled as an Adobe Acrobat PDF file. The Acrobat Distiller and Acrobat Reader are not included with Corel VENTURA.

Syntax	Description
<b>.FileName</b>	Specifies the name and path of the output file.
<b>.PrinterName</b>	Specifies the name of the PostScript printer driver to use. The default is the current printer.

### Example

```
.FilePublishAsAcrobat "C:\WIN95\Desktop\MyFile.prn", "HP LaserJet 4P/4MP PostScript"
```

This example publishes the current publication to a file named MYFILE.PRN which can then be distilled as a PDF file.

---

**{button ,AL('vent\_file;;;',0,"Defaultoverview",)} Related Topics**

## .FilePublishAsBarista (VENTURA)

**.FilePublishAsBarista** **.FileName=string**, **.MultipleFile=Boolean**, **.GraphicFormat=long**, **.CurrentPage=Boolean**, **.EmbedFonts=Boolean**

This command publishes the active publication to a Corel Barista file (an [HTML](#) file) for viewing in a [Java](#) -enabled [Web browser](#) such as Netscape Navigator 2.0 or later.

Syntax	Description
<b>.FileName</b>	Specifies the filename and path the Corel Barista file will be saved to. If the path information is not included, the active folder is used.
<b>.MultipleFile</b>	Specifies whether to publish the publication as a single file or to publish each page in the publication as a separate file. Set to TRUE (-1) to publish each page in the publication as a separate file; otherwise, set to FALSE (0) which is the default if omitted.
<b>.GraphicFormat</b>	Specifies the file format COREL VENTURA converts pictures to: 0      GIF (default if omitted) 1      JPEG GIF graphics generally display faster than JPEG graphics, whereas JPEG compresses the picture to a smaller size.
<b>.CurrentPage</b>	Specifies whether to publish the entire publication or the active page. Set to TRUE (-1) to publish the current page; otherwise, set to FALSE (0) which is the default if omitted.
<b>.EmbedFonts</b>	Specifies whether to have the fonts used in published publication saved as .PFR files (TrueDoc fonts). Then, if the required fonts are not available on a user's computer, the .PFR files are downloaded to supply the fonts. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0), which is the default if omitted.

### Note



This command cannot be recorded.



This command corresponds to the Publish As Barista dialog box. Click File, Publish As, Barista.



The [.FilePublishAsHTML](#) command can also be used to publish COREL VENTURA publications.



This command uses the current COREL VENTURA settings in determining whether to display the published publication in installed WEB browser.

### Example

```
.FileOpen "C:\VENTURA\MYFILES\TEST1.VP"  
.FilePublishAsBarista "c:\Java\My_WWW.htm", FALSE, TRUE, FALSE, TRUE
```

The above example opens a COREL VENTURA publication named TEST1.VP. The second line publishes the open publication as a Corel Barista publication (MY\_WWW.HTM). TrueDoc fonts are imbedded in the publication, and pictures are converted to the JPEG format in the above example.

---

{button ,AL('vent\_file;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FilePublishAsCSS (VENTURA)

**.FilePublishAsBarista** **.FileName**=*string*, **.BodyElementTag**=*string*, **.HTMLTagOnly**=*Boolean*, **.DifferenceInBody**=*Boolean*

This command publishes the active publication as a cascading style sheet.

Syntax	Description
<b>.FileName</b>	Specifies the filename and path of the cascading style sheet. If the path information is not included, the active folder is used.
<b>.BodyElementTag</b>	Specifies the Corel VENTURA paragraph tag to use to represent the <BODY> style in the cascading style sheet.
<b>.HTMLTagOnly</b>	Specifies whether to export only tags that are HTML elements. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0) which is the default if omitted.
<b>.DifferenceInBody</b>	Specifies whether to specify only styles that are different from the <BODY> style.. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). The default is TRUE if omitted.

### Note



This command was introduced in Corel VENTURA 8.



The **.FilePublishAsHTML** command can also be used to publish Internet documents.

### Example

```
.FileOpen "C:\VENTURA\MYFILES\TEST1.VP"  
.FilePublishAsCSS "c:\corel\Publication1.css", "Body Text",
```

The above example opens a Corel VENTURA publication named TEST1.VP. The second line publishes the open publication as a CSS file.

---

{button ,AL('vent\_file;;;','0,"Defaultoverview",,)} [Related Topics](#)

## **.FilePublishAsEnvoy (VENTURA)**

### **.FilePublishAsEnvoy**

This command is only available in Corel VENTURA 7. Use one of the following commands to publish:

- [.FilePublishAsAcrobat](#)
- [.FilePublishAsBarista](#)
- [.FilePublishAsCSS](#)
- [.FilePublishAsHTML](#)

---

`{button ,AL('vent_file;;;','0',"Defaultoverview"),}` [Related Topics](#)

## .FilePublishAsHTML (VENTURA)

**.FilePublishAsHTML** **.FileName**=*string*, **.Range**=*long*, **.AutoHTMLTags**=*Boolean*

This command publishes the active publication to an HTML file for viewing in a Web browser such as Netscape Navigator 2.0 or later.

Syntax	Description
<b>.FileName</b>	Specifies the name of the published publication. If the path information is not included, the active folder is used.
<b>.Range</b>	Specifies the range to publish: 0 Entire publication 1 Current chapter (default if omitted) 2 Current page
<b>.AutoHTMLTags</b>	Specifies whether to have COREL VENTURA convert the publication into the closest HTML approximation. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0), which is the default if omitted.

### Note



This command cannot be recorded.



This command corresponds to the Publish As HTML dialog box. Click File, Publish As, HTML. However, this command cannot customize the HTML conversion by specifying how text formatting in the publication is translated. Click Setup in the dialog box for more information.



The **.FilePublishAsBarista** command can also be used to publish Corel VENTURA publications.



This command uses the current Corel VENTURA settings in determining whether to display the published publication in installed WEB browser.

### Example

```
.FileOpen "C:\VENTURA\MYFILES\TEST1.VP"  
.FilePublishAsHTML "c:\Web\My_WWW.htm", 0, TRUE
```

The above example opens a Corel VENTURA publication named TEST1.VP. The second line publishes the open publication as an HTML file (MY\_WWW.HTM). The AutoHTML tags option is also enabled.

---

{button ,AL('vent\_file;;;','0',"Defaultoverview"),} Related Topics

## .FileRemove (VENTURA)

**.FileRemove** *.FileName=string, .FromFrame=Boolean, .FromChapter=Boolean, .GraphicFile=Boolean, .ChapterName=string*

This command removes imported text and picture files from the active frame or chapter.

Syntax	Description
<b>.FileName</b>	<p>Specifies the name of the file to remove.</p> <p>If the text file is Export On Save enabled, use the entire path and extension of the text file (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used).</p> <p>If the picture file is embedded, use the file name that appears in the VENTURA Navigator or File List. If the picture file is external, use the picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.</p> <p>This parameter was modified in Corel VENTURA 8.</p>
<b>.FromFrame</b>	<p>Specifies whether to remove the file from the frame. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0).</p>
<b>.FromChapter</b>	<p>Specifies whether to remove the file from the chapter. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0). If omitted, the default is FALSE.</p>
<b>.GraphicFile</b>	<p>Specifies whether to remove a picture or text file. Set to TRUE (-1) to remove a picture file. Set to FALSE (0) to remove a text file.</p>
<b>.ChapterName</b>	<p>Specifies the name of the chapter to remove the file from. If omitted, the active chapter is used.</p>

### Example

```
.FileRemove "Old Text" , 0, -1, 0, "Old section"
```

The above example removes the Old Text file from the Old section chapter.

---

`{button ,AL('vent_file;;;;',0,"Defaultoverview"),}` [Related Topics](#)

## .FileRename (VENTURA)

**.FileRename** *.FileName=string*, *.NewName=string*, *.GraphicFile=Boolean*, *.ChapterName=string*

This command renames a specified text or picture file in the active chapter.

Syntax	Description
<b>.FileName</b>	<p>Specifies the name of the file to rename.</p> <p>If the text file is Export On Save enabled, use the entire path and extension of the text file (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used).</p> <p>If the picture file is embedded, use the file name that appears in the VENTURA Navigator or File List. If the picture file is external, use the picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.</p> <p>This parameter was modified in Corel VENTURA 8.</p>
<b>.NewName</b>	<p>Specifies the new name for the specified file.</p>
<b>.GraphicFile</b>	<p>Specifies whether to rename a picture or text file. Set to TRUE (-1) to rename a picture file. Set to FALSE (0) to rename a text file. If omitted, the default is FALSE.</p>
<b>.ChapterName</b>	<p>Specifies the name of the chapter where the file to be renamed resides (case sensitive). If omitted, the active chapter is used.</p>

### Example

```
.FileRename "oldname" , "newname"
```

The above example renames OLDNAME to NEWNAME.

---

{button ,AL('vent\_file;;;','0',"Defaultoverview"),} [Related Topics](#)

## .FileRenameTextFile (VENTURA)

**.FileRenameTextFile** *.FileName=string, .NewName=string, .FilterKey=long, .ExportOnSave=Boolean, .ChapterName=string*

This command renames a specified text file in the active chapter.

Syntax	Description
<b>.FileName</b>	Specifies the name of the file to remove. If the text file is Export On Save enabled, use the entire path and extension of the text file (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used). This parameter was modified in Corel VENTURA 8.
<b>.NewName</b>	Specifies the new name for the specified file.
<b>.FilterKey</b>	Specifies the filter to be used for export: 11 Rich Text Format(.RTF) 51 ANSI text (.TXT) 52 ASCII text (.TXT) 53 ASCII 8-bit text (.TXT)
<b>.ExportOnSave</b>	Specifies whether the text file is linked or embedded. Set to TRUE (-1) to link the text file. Set to FALSE (0) to embed the text file. If this parameter is set to TRUE, the <b>.TextFileGetAt</b> function will return the file name and extension.
<b>.ChapterName</b>	Specifies the name of the chapter where the file to be renamed resides (case sensitive). If omitted, the active chapter is used.

### Note

**I** This command was introduced in Corel VENTURA 8.

### Example

```
.FileRenameTextFile "oldname" , "newname"
```

The above example renames OLDNAME to NEWNAME.

---

{button ,AL('vent\_file;;;','0',"Defaultoverview"),} [Related Topics](#)



## **.FileRevert (VENTURA)**

### **.FileRevert**

This command resets the active publication to it's last saved version. All edits are lost when this command is executed.

#### **Notes**



This command was introduced in Corel VENTURA 8.



This command corresponds to clicking File, Revert.

#### **Example**

.FileRevert

---

{button ,AL(^vent\_file;;;','0,"Defaultoverview",,)} Related Topics

## .FileSave (VENTURA)

**.FileSave** .SaveAsName=*string*

This command saves the active Corel VENTURA publication.

Syntax	Description
.SaveAsName	If publication had not been previously named, this parameter can be used to specify the publication name. If omitted, the Save As dialog box opens.

### Note



This command corresponds to the Save command in the File menu in Corel VENTURA. Click File, Save.



The **.SaveAsName** parameter was introduced in Corel VENTURA 8.

### Example

.FileSave

---

{button ,AL(^vent\_file;;;',0,"Defaultoverview"),} Related Topics

## .FileSaveAs (VENTURA)

**.FileSaveAs** **.FileName=string**, **Overwrite=long**

This command saves the active publication under a new name or in a different location.

Syntax	Description
<b>.FileName</b>	Specifies the name of the publication. If the path information is not included, the active folder is used. If a publication with the name specified in <b>.FileName</b> already exists in the folder, it will be overwritten without prompting you.
<b>.Overwrite</b>	Specifies an option if the file already exists. 0 Do not overwrite 1 Overwrite (default if omitted) 2 Prompt user

### Note



This command corresponds to the Save As dialog box. Click File, Save As.



By default, COREL VENTURA makes backup copies of publications each time you save them, and at regular intervals while you're working on them. You can turn the backup feature off using the Tools, Options command, although it is best to leave them on to save yourself from losing work.



The **.Overwrite** parameter was introduced in Corel VENTURA 8.

### Example

```
.FileOpen "C:\VENTURA\MYFILES\TEST1.VP"  
.FileSaveAs "X:\MyBox\TEST1.VP"
```

The above example opens a Corel VENTURA publication named TEST1.VP. The open publication is then saved to the X drive.

---

{button ,AL('vent\_file;;;','0',"Defaultoverview"),} [Related Topics](#)

## .FileSavePageAsEPS (VENTURA)

**.FileSavePageAsEPS** *.FileName=string*

This command saves the current page as an EPS file.

Syntax	Description
<b>.FileName</b>	Specifies the name of the published publication. If the path information is not included, the active folder is used.

### Note



This command was introduced in Corel VENTURA 8.



This command corresponds to clicking File, Export Page as EPS.

### Example

```
.FileOpen "C:\VENTURA\MYFILES\TEST1.VP"  
.FileSavePageAsEPS "c:\My_EPS.EPS"
```

The above example opens a Corel VENTURA publication named TEST1.VP. The second line saves the first page of the publication as an EPS file named MY\_EPS.

---

{button ,AL('vent\_file;;;','0',"Defaultoverview",,)} Related Topics

## .FillFountain (VENTURA)

**.FillFountain** .Type=*long*, .CenterX=*long*, .CenterY=*long*, .Angle=*long*, .Steps=*long*, .Padding=*long*, .Blend=*long*, .Rate=*long*

This command, along with the **.FillFountainColor** commands, sets the fountain fill attributes of a selected object. Objects include graphics, text, and frames.

This command must be followed by a contiguous block of **.FillFountainColor** commands. Each **.FillFountainColor** command sets color attributes for the fountain fill color blend. The number of **.FillFountainColor** commands required is dependent of the number of colors in the color blend. For example, a two-color blend uses 2 **.FillFountainColor** commands. See the example below for more information.

Syntax	Description
.Type	Specifies the fountain fill type: 0     Linear (default if omitted) 1     Radial 2     Conical 3     Square
.CenterX	If <b>.Type</b> is not set to Linear, specifies a horizontal center for the selected object's fountain fill. Negative values shift the fill to the left; positive values shift the fill to the right. This parameter uses values between -100 and 100, inclusive.
.CenterY	If <b>.Type</b> is not set to Linear, specifies a vertical center for the selected object's fountain fill. Negative values shift the fill down; positive values shift the fill up. This parameter uses values between -100 and 100, inclusive.
.Angle	Specifies the angle of linear, conical, and square fountain fills in tenths of a degree (1 degree equals 10 tenths of a degree). Positive values rotate the fill counter-clockwise; negative values rotate the fill clockwise.
.Steps	Specifies the number of bands used to display and print the fountain fill. For a smoother gradation, increase the number of steps. Valid values range from 2 to 256, inclusive.
.Padding	Specifies the amount to increase the edge padding as a percentage of the selected object size. Valid values range from 0 to 45, inclusive. The <b>.Padding</b> parameter is ignored with conical fills.
.Blend	In a two-color fountain fill, specifies the method to determine the intermediate fill colors: 0     Straight-line across the color wheel 1     Clockwise path around the color wheel 2     Counter-clockwise path around the color wheel
.Rate	Specifies the mid-point in a two-color fountain fill. Valid values range from 1 to 99, inclusive.

### Note



This command corresponds to the Fountain Fill dialog box in Corel VENTURA. Click the



button.

### Example

See the example in the **.FillFountainColor** command.

---


{button ,AL("VENTURA\_Fill\_Menu;;;;','0,"Defaultoverview",)} Related Topics

## .FillFountainColor (VENTURA)

**.FillFountainColor .Position=*long*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long***

This command, along with the **.FillFountain** commands, sets the fountain fill attributes of a selected object. Objects include graphics, text, and frames.

This command is always found in a contiguous block of two or more **.FillFountainColor** commands, and the block must be preceded by the **.FillFountain** command. Each **.FillFountainColor** command sets color attributes for the fountain fill color blend. The number of **.FillFountainColor** commands required is dependent on the number of colors in the color blend. For example, a two-color blend uses 2 **.FillFountainColor** commands. See the example for more information.

Syntax	Description
<b>.Position</b>	Specifies the position of the color in the color blend. The first <b>.FillFountainColor</b> command in the contiguous block must have this parameter set to 0; the last command in the block must have this parameter set to 100. Valid values range from 0 to 100, inclusive.
<b>.ColorModel</b>	<p>Specifies the color model to use for the specified position in the color blend:</p> <ol style="list-style-type: none"><li>1 <a href="#">PANTONE Spot Colors</a></li><li>2 <a href="#">CMYK</a></li><li>3 <a href="#">CMYK255</a></li><li>4 <a href="#">CMY</a></li><li>5 <a href="#">RGB</a> (default if omitted)</li><li>6 <a href="#">HSB</a></li><li>7 <a href="#">HLS</a></li><li>8 <a href="#">Black and White</a></li><li>9 <a href="#">Grayscale</a></li><li>11 <a href="#">YIQ</a></li><li>12 <a href="#">L*a*b*</a></li><li>14 <a href="#">PANTONE Hexachrome</a></li></ol> <p>If the <b>.FillFountainColor</b> command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.</p>
<b>.Color1</b>	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color3</b>	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
<b>.Color4</b>	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note



This command corresponds to the Fountain Fill dialog box in Corel VENTURA. Click the



button.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.FillFountain 3, 5, 10, 200, 107, 4, 1, 65
.FillFountainColor 0, 2, 0, 100, 100, 0 'first color -- red
.FillFountainColor 100, 2, 100, 100, 0, 0 'second color -- blue
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The ellipse is then filled with a two-color fountain fill (red to blue). The **.LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.FillFountain 0, 0, 0, 200, 107, 0, 3, 50
.FillFountainColor 0, 2, 0, 100, 100, 0 'first color -- red
```

```
.FillFountainColor 15, 2, 100, 0, 100, 0    'second color -- green  
.FillFountainColor 30, 2, 100, 100, 0, 0    'third color -- blue  
.FillFountainColor 60, 2, 0, 0, 100, 0      'fourth color -- yellow  
.FillFountainColor 100, 2, 0, 0, 0, 100     'fifth color -- black
```

In this example, an ellipse is filled with a five-color fountain fill (red, green, blue, yellow, black). The colors are positioned at 0, 15, 30, 60, and 100.

---

**{button ,AL("VENTURA\_Fill\_Menu;;;;','0,"Defaultoverview",)} Related Topics**





## .FillFountainColorGet (VENTURA)

**.FillFountainColorGet** .Index=*long*, .Position=*long*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This function returns the fountain fill color and position attributes of a selected object. Objects include graphics, text, and frames. To return other fountain fill attributes, see the [.FillFountainGet](#) command.

Syntax	Description
.Index	Specifies the color in the color blend to return attributes for. The first color in the color blend is 0, the second color is 1, and so on.
.Position	Specifies the numeric variable that is assigned the position of the color in the color blend. Valid values range from 0 to 100, inclusive.
.ColorModel	Specifies the numeric variable that is assigned a return value corresponding to the specified color blend index's color model: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
.Color1	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color3	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color4	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

### Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  If more than one object is selected, this function returns the attributes of the first selected object.
-  If the selected object is not filled with a fountain fill (for example, a uniform color), 0 is assigned to each variable used in the function.
-  This command cannot be recorded.

### Example

```
.SelectObjectAt 2.5*M_INCH, 3*M_INCH  
.FillFountainGet FFType, , FFAngle, FFSteps, FFPad, , FFNum
```

The above example selects an object on the active page. The selected object's fountain fill attributes are then assigned. The number of colors in the color blend is assigned to the **FFNum** variable. The **FFNum** variable's contents can then be used to determine the position and color attributes of the last color in the color blend.

```
.FillFountainColorGet FFNum, FFPos, FFModel, C1, C2, C3, C4
```

---

{button ,AL('VENTURA\_Fill\_Menu;;;;','0',"Defaultoverview",)} [Related Topics](#)



# .FillFountainColorRemove (VENTURA)

**.FillFountainColorRemove .Position=*long***

This command removes a specified color from a fountain fill's color blend on a selected object.

Syntax	Description
<b>.Position</b>	Specifies the position of the color to remove. This position must exist in fountain fill's color blend on the selected object.

**Note**

**I**

This command cannot be recorded.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.FillFountain 0, 0, 0, 200, 107, 0, 3, 50
.FillFountainColor 0, 2, 0, 100, 100, 0      'first color -- red
.FillFountainColor 15, 2, 100, 0, 100, 0    'second color -- green
.FillFountainColor 30, 2, 100, 100, 0, 0    'third color -- blue
.FillFountainColor 60, 2, 0, 0, 100, 0      'fourth color -- yellow
.FillFountainColor 100, 2, 0, 0, 0, 100     'fifth color -- black
,
.EditDuplicate
.FillFountainColorRemove 15
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The ellipse is filled with a five-color fountain fill (red, green, blue, yellow, black). The colors are positioned at 0, 15, 30, 60, and 100. The ellipse is then duplicated, and the duplicate has the green color (15) removed from its color blend.





# .FillFountainGet (VENTURA)

**.FillFountainGet** .Type=*long*, .CenterX=*long*, .CenterY=*long*, .Angle=*long*, .Steps=*long*, .Padding=*long*, .Blend=*long*, .Rate=*long*, .NumOfColors=*long*

This function returns the fountain fill attributes of a selected object. Objects include graphics, text, and frames. To return other fountain fill attributes, see the **.FillFountainColorGet** command.

Syntax	Description
.Type	Specifies the variable that is assigned the fountain fill type: 0     Linear 1     Radial 2     Conical 3     Squarer
.CenterX	Specifies the variable that is assigned the horizontal center for the selected object's fountain fill. This parameter uses values between -100 and 100, inclusive.
.CenterY	Specifies the variable that is assigned the vertical center for the selected object's fountain fill. This parameter uses values between -100 and 100, inclusive.
.Angle	Specifies the variable that is assigned the angle of linear, conical, and square fountain fills, in tenths of a degree (1 degree equals 10 tenths of a degree). Positive values rotate the fill counter-clockwise; negative values rotate the fill clockwise.
.Steps	Specifies the variable that is assigned the number of bands used to display and print the fountain fill. Valid values range from 2 to 256, inclusive.
.Padding	Specifies the variable that is assigned the edge padding amount. Valid values range from 0 to 45, inclusive.
.Blend	For a two-color fountain fill, specifies the variable that is assigned the intermediate fill colors method: 0     Straight-line across the color wheel 1     Clockwise path around the color wheel 2     Counter-clockwise path around the color wheel
.Rate	For two-color fountain fill, specifies the variable that is assigned the mid-point.
.NumOfColors	Specifies the variable that is assigned the number of colors in the color blend.

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  If more than one object is selected, this function returns the attributes of the first selected object.
-  If the selected object is not filled with a fountain fill (for example, a uniform color), 0 is assigned to each variable used in the function.
-  This command cannot be recorded.

## Example

.SelectObjectAt 2.5\*M\_INCH, 3\*M\_INCH  
.FillFountainGet FFType, , ,FFAngle, FFSteps, FFPad, , , FFNum

The above example selects an object on the active page. The selected object's fountain fill attributes are then assigned. The number of colors in the color blend is assigned to the **FFNum** variable. The **FFNum** variable's contents can then be used to determine the position and color attributes of the last color in the color blend.

.FillFountainColorGet FFNum, FFPos, FFModel, C1, C2, C3, C4

{button ,AL('VENTURA\_Fill\_Menu;;;;','0,'Defaultoverview',)} Related Topics

## .FillFullColor (VENTURA)

**.FillFullColor** **.FileName=string**, **.TileWidth=long**, **.TileHeight=long**, **.TileOffsetX=long**, **.TileOffsetY=long**, **.RowColOffset=long**, **.Column=Boolean**, **.SeamlessTile=Boolean**, **.ScaleWithObject=Boolean**

This command sets a full-color bitmap pattern fill for a selected object. Objects include graphics, text, and frames.

Syntax	Description
<b>.FileName</b>	Specifies the name and path of the imported bitmap to use as a fill.  Do not record this command because this parameter will be filled with the string "Unkown".
<b>.TileWidth</b>	Specifies the width of the bitmap pattern. If the <b>.ScaleWithObject</b> parameter is set to TRUE (-1), it is expressed as a percentage of the width of the selected object. If the <b>.ScaleWithObject</b> parameter is set to FALSE (0), it is expressed as an absolute width in tenths of a micron.
<b>.TileHeight</b>	Specifies the height of the bitmap pattern. If the <b>.ScaleWithObject</b> parameter is set to TRUE (-1), it is expressed as a percentage of the height of the selected object. If the <b>.ScaleWithObject</b> parameter is set to FALSE (0), it is expressed as an absolute height in tenths of a micron.
<b>.TileOffsetX</b>	Specifies the horizontal position of the first pattern tile relative to the left side of the selected object. This parameter is expressed as a percentage of the width of the bitmap pattern.
<b>.TileOffsetY</b>	Specifies the vertical position of the first pattern tile relative to the top of the selected object. This parameter is expressed as a percentage of the height of the bitmap pattern.
<b>.RowColOffset</b>	Specifies the amount to shift the alternating rows of tiles or columns of tiles as percentage of the bitmap pattern. Use the <b>.Column</b> parameter to specify shifting by rows or columns.
<b>.Column</b>	Specifies whether to shift alternating rows of tiles or columns of tiles. Set to TRUE (-1) to shift columns. Set to FALSE (0) to shift rows which is the default if omitted.
<b>.SeamlessTile</b>	Specifies whether to enable seamless tiling which alters the pattern tiles so that they appear to be continuous. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0). If omitted, the default is TRUE (-1).
<b>.ScaleWithObject</b>	Specifies whether to use absolute and proportional pattern tile sizes. Set to TRUE (-1) to use proportional pattern tile sizes. Set to FALSE (0) to use absolute pattern tile sizes which is the default if omitted. This parameter effects the units used in the <b>.TileWidth</b> and <b>.TileHeight</b> parameters.

### Note



This command corresponds to clicking Import button in the Full-Color Bitmap Pattern dialog box in Corel VENTURA. Click the



button. This command cannot use a predefined bitmap as you can with the Full-Color Bitmap Pattern dialog box.



Do not record this command because the first parameter will be filled with the string "Unkown".

### Example

**.GraphicFirst**  
**.FillFullColor** "c:\Colors.bmp", 63500, 63500, 0, 0, 0, FALSE, FALSE, FALSE

The above example fills the first graphic on the active page with the COLORS.BMP bitmap

---

**{button ,AL('VENTURA\_Fill\_Menu;;;;','0','Defaultoverview',)} Related Topics**

## .FillFullColorGet (VENTURA)

**.FillFullColorGet** .FileName=*string*, .TileWidth=*long*, .TileHeight=*long*, .TileOffsetX=*long*, .TileOffsetY=*long*, .RowColOffset=*long*, .Column=*Boolean*, .SeamlessTile=*Boolean*, .ScaleWithObject=*Boolean*

This function returns the full-color bitmap pattern fill settings of a selected object. Objects include graphics, text, and frames.

Syntax	Description
.FileName	Omit this parameter since it currently cannot be used to hold a return value.
.TileWidth	Specifies the numeric variable that is assigned the width of the bitmap pattern. If the <b>.ScaleWithObject</b> parameter is set to TRUE (-1), this is expressed as a percentage of the width of the selected object. If the <b>.ScaleWithObject</b> parameter is set to FALSE (0), this is expressed as an absolute width in tenths of a micron.
.TileHeight	Specifies the numeric variable that is assigned the height of the bitmap pattern. If the <b>.ScaleWithObject</b> parameter is set to TRUE (-1), this is expressed as a percentage of the height of the selected object. If the <b>.ScaleWithObject</b> parameter is set to FALSE (0), this is expressed as an absolute height in tenths of a micron.
.TileOffsetX	Specifies the numeric variable that is assigned the horizontal position of the first pattern tile relative to the left side of the selected object. The assigned value is expressed as a percentage of the width of the bitmap pattern.
.TileOffsetY	Specifies the numeric variable that is assigned the vertical position of the first pattern tile relative to the top of the selected object. The assigned value is expressed as a percentage of the height of the bitmap pattern.
.RowColOffset	Specifies the numeric variable that is assigned the amount to shift the alternating rows of tiles or columns of tiles. The assigned value is expressed as percentage of the bitmap pattern.
.Column	Specifies the numeric variable that is assigned a value corresponding to whether rows of tiles or columns of tiles are shifted: TRUE (-1) for columns; FALSE (0) for rows.
.SeamlessTile	Specifies the numeric variable that is assigned a value corresponding to whether seamless tiling is enabled: TRUE (-1) if enabled; otherwise FALSE (0).
.ScaleWithObject	Specifies the numeric variable that is assigned a value corresponding to whether absolute and proportional pattern tile sizes are used: TRUE (-1) for proportional pattern tile sizes; FALSE (0) for absolute pattern tile sizes.

### Note

- 1** The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
- 1** If more than one object is selected, this function returns the attributes of the first selected object.
- 1** If the selected object is not filled with a full-color bitmap pattern fill (for example, a fountain fill), 0 is assigned to each variable used in the function.
- 1** This command cannot be recorded.

### Example

```
.GraphicFirst
.FillFull "Unknown", 63500, 63500, 0, 0, 0, FALSE, FALSE, FALSE
.FillFullColorGet , TW&, TH&, TO&, RO&, COL&, ST&, SWO&
```

The above example selects the first graphic on the active page and returns its properties to the specified variables.

---

{button ,AL('VENTURA\_Fill\_Menu;;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.FillNoFill (VENTURA)**

### **.FillNoFill**

This command removes the fill from selected objects.

#### **Note**



This command corresponds to clicking the



button.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
```

```
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
```

```
.FillUniform 2, 40, 0, 20, 60
```

```
.EditDuplicate
```

```
.FillNoFill
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame. The ellipse is then filled with a dark green color using the CMYK color model. The ellipse is duplicated and the fill is then removed from the duplicate ellipse.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Fill\_Menu;;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.FillOverprint (VENTURA)**

**.FillOverprint .Overprint=Boolean,**

This command sets the overprint option for a selected object's fill. Objects include graphics, text, and frames. Enabling the overprint option allows the fill to print over the background color. This option is relevant only when printing color separations.

Syntax	Description
<b>.Overprint</b>	Specifies whether to enable the overprint option. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0) which is the default if omitted.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.FillUniform 2, 40, 0, 20, 60
.FillOverprint TRUE
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame. The ellipse is then filled with a dark green color using the CMYK color model. The ellipse's fill then has the overprint option applied to it.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

**{button ,AL("VENTURA\_Fill\_Menu;;;','0,"Defaultoverview",)} Related Topics**

**.FillOverprintGet (VENTURA)**

ReturnValue = .FillOverprintGet ( )

This function returns the overprint fill setting of a selected object. Objects include graphics, text, and frames.

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a value corresponding to whether the overprint option is enabled: TRUE (-1) if enabled; otherwise FALSE (0).

**Note**

- 1** If more than one object is selected, this function returns the attributes of the first selected object.
- 1** This command cannot be recorded.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FillUniformGet cm&, c1&, c2&, c3&, c4&
OPoption = .FillOverprintGet ( )
```

The above example draws a rectangle and an ellipse. The rectangle is then selected using the .SelectObjectAt command. The .FillUniformGet function returns the uniform color values for the rectangle to the variables **cm**, **c1**, **c2**, **c3**, and **c4**. The .FillOverprintGet function returns the overprint status to the **OPoption** variable.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.


**.FillPostscript (VENTURA)**


**.FillPostscript** *.Name=Fi string, .NumOfParams=long, .Value1=long, .Value2=long, .Value3=long, .Value4=long, .Value5=long*

This command sets a PostScript texture fill for a selected object. Objects include graphics, text, and frames.

Syntax	Description
<b>.Name</b>	Specifies the name of the PostScript texture fill to apply. The name must be preceded by an <b>F/</b> . For a listing of available Postscript fills, see the PostScript Texture dialog box.
<b>.NumOfParams</b>	Specifies the number of setting parameters available to use with the specified PostScript texture fill (a value between 1 and 5, inclusive). Refer to the PostScript Texture dialog box to determine the number of parameters for a specified PostScript fill.
<b>.Value1</b>	Specifies the first parameter for the selected PostScript Fill. This parameter varies depending on the fill selected. Refer to the PostScript Texture dialog box for fill parameter specifics.
<b>.Value2</b>	Specifies the second parameter for the selected PostScript Fill. This parameter varies depending on the fill selected. Refer to the PostScript Texture dialog box for fill parameter specifics. If the parameter is not used with the specified fill, omit it.
<b>.Value3</b>	Specifies the third parameter for the selected PostScript Fill. This parameter varies depending on the fill selected. Refer to the PostScript Texture dialog box for fill parameter specifics. If the parameter is not used with the specified fill, omit it.
<b>.Value4</b>	Specifies the fourth parameter for the selected PostScript Fill. This parameter varies depending on the fill selected. Refer to the PostScript Texture dialog box for fill parameter specifics. If the parameter is not used with the specified fill, omit it.
<b>.Value5</b>	Specifies the fifth parameter for the selected PostScript Fill. This parameter varies depending on the fill selected. Refer to the PostScript Texture dialog box for fill parameter specifics. If the parameter is not used with the specified fill, omit it.

**Note**



This command corresponds to the PostScript Fill dialog box in Corel VENTURA. Click the  button.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.FillPostscript "F/DNA", 5, 9, 3, 102, -3, 90
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame. The ellipse is then filled with the DNA texture.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Fill\_Menu;;;;','0,"Defaultoverview",')} Related Topics









# .FillPostscriptGet (VENTURA)

**.FillPostscriptGet** .Name=*string*, .NumOfParams=*long*, .Value1=*long*, .Value2=*long*, .Value3=*long*, .Value4=*long*, .Value5=*long*

This function returns the PostScript text fill parameters of a selected object. Objects include graphics, text, and frames.

Syntax	Description
.Name	Specifies the variable that is assigned the name of the PostScript texture fill. This name is preceded by <b>F</b> .
.NumOfParams	Specifies the variable that is assigned the number of setting parameters available to use with the PostScript texture fill.
.Value1	Specifies the variable that is assigned the first parameter setting for the PostScript fill. This parameter varies depending on the PostScript fill being used. Refer to the PostScript Texture dialog box for fill parameter specifics.
.Value2	Specifies the variable that is assigned the second parameter setting for the PostScript fill. This parameter varies depending on the PostScript fill being used. Refer to the PostScript Texture dialog box for fill parameter specifics. If the parameter is not used in PostScript fill, 0 is returned.
.Value3	Specifies the variable that is assigned the third parameter setting for the PostScript fill. This parameter varies depending on the PostScript fill being used. Refer to the PostScript Texture dialog box for fill parameter specifics. If the parameter is not used in PostScript fill, 0 is returned.
.Value4	Specifies the variable that is assigned the fourth parameter setting for the PostScript fill. This parameter varies depending on the PostScript fill being used. Refer to the PostScript Texture dialog box for fill parameter specifics. If the parameter is not used in PostScript fill, 0 is returned.
.Value5	Specifies the variable that is assigned the fifth parameter setting for the PostScript fill. This parameter varies depending on the PostScript fill being used. Refer to the PostScript Texture dialog box for fill parameter specifics. If the parameter is not used in PostScript fill, 0 is returned.

## Note

-  The parameters in this function correspond to the dialog box controls in the PostScript Fill dialog box (click ).
-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  If more than one object is selected, this function returns the attributes of the first selected object.
-  If the selected object is not filled with a PostScript fill (for example, a fountain fill), 0 is assigned to each variable used in the function.
-  This command cannot be recorded.

## Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FillPostScriptGet PSName, ParNum, P1, P2, P3, P4, P5
```

The above example draws a rectangle and an ellipse. The rectangle is then selected using the .SelectObjectAt command. The **.FillPostscriptGet** function then returns the texture fill values for the rectangle to the variables **PSName**, **ParNum**, **P1**, **P2**, **P3**, **P4**, and **P5**.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

# .FillTexture (VENTURA)

**.FillTexture** .Library=*string*, .Name=*string*, .Style=*string*

This command sets a bitmap texture fill for a selected object. Objects include graphics, text, and frames.

Syntax	Description
.Library	Specifies the texture library that contains the texture to apply. If not specified, the default is Styles.
.Name	Specifies the name of the texture to apply.
.Style	Specifies the style name of the name of the texture to apply. This name corresponds to the name listed in the Style Name group box in the Texture Fill dialog box (see the Note section below for more information).

## Note



You cannot use this command to change a texture's settings.



This command corresponds to the Texture Fill dialog box in Corel VENTURA. Click the



button.

## Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.FillTexture "Samples 5", "Blocks-Rainbow", "Cdr5:Surfaces-Rainbow"
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame. The ellipse is then filled with the Blocks-Rainbow texture.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Fill\_Menu;;;;";0,"Defaultoverview"),} Related Topics







# .FillTextureGet (VENTURA)

**.FillTextureGet** .Library=*string*, .Name=*string*, .Style=*string*

This function returns the texture fill parameters of a selected object. Objects include graphics, text, and frames.

Syntax	Description
.Library	Specifies the variable that is assigned the texture library name that contains the texture.
.Name	Specifies the variable that is assigned the name of the texture.
.Style	Specifies the variable that is assigned the texture style name.

### Note

-  The parameters in this function correspond to the dialog box controls in the Texture Fill dialog box (click ,).
-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  If more than one object is selected, this function returns the attributes of the first selected object.
-  If the selected object is not filled with a texture fill (for example, a fountain fill), 0 is assigned to each variable used in the function.
-  This command cannot be recorded.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FillTextureGet TexLib, TexName, TexStyle
```

The above example draws a rectangle and an ellipse. The rectangle is then selected using the .SelectObjectAt command. The .FillTextureGet function then returns the texture fill values for the rectangle to the variables **TexLib**, **TexName**, and **TexStyle**.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

## .FillTwoColor (VENTURA)

**.FillTwoColor** **.FileName=string**, **.BackModel=long**, **.BackColor1=long**, **.BackColor2=long**, **.BackColor3=long**, **.BackColor4=long**, **.FrontModel=long**, **.FrontColor1=long**, **.FrontColor2=long**, **.FrontColor3=long**, **.FrontColor4=long**, **.TileWidth=long**, **.TileHeight=long**, **.TileOffsetX=long**, **.TileOffsetY=long**, **.RowColOffset=long**, **.Column=Boolean**, **.SeamlessTile=Boolean**, **.ScaleWithObject=Boolean**

This command sets an imported two-color bitmap pattern fill for a selected object. Objects include graphics, text, and frames.

Syntax	Description
<b>.FileName</b>	Specifies the name and path of the imported bitmap to use as a fill.  Do not record this command because this parameter will be filled with the string "Unkown".
<b>.BackModel</b>	Specifies the color model to use for the background color: <div><div>1</div><div><a href="#">PANTONE Spot Colors</a></div></div> <div><div>2</div><div><a href="#">CMYK</a></div></div> <div><div>3</div><div><a href="#">CMYK255</a></div></div> <div><div>4</div><div><a href="#">CMY</a></div></div> <div><div>5</div><div>RGB (default if omitted)</div></div> <div><div>6</div><div><a href="#">HSB</a></div></div> <div><div>7</div><div><a href="#">HLS</a></div></div> <div><div>8</div><div><a href="#">Black and White</a></div></div> <div><div>9</div><div><a href="#">Grayscale</a></div></div> <div><div>11</div><div><a href="#">YIQ</a></div></div> <div><div>12</div><div><a href="#">L*a*b*</a></div></div> <div><div>14</div><div><a href="#">PANTONE Hexachrome</a></div></div>

.TileOffsetY	Specifies the vertical position of the first pattern tile relative to the top of the selected object. The parameter is expressed as a percentage of the height of the bitmap pattern
.RowColOffset	Specifies the amount to shift the alternating rows of tiles or columns of tiles as percentage of the bitmap pattern. Use the <b>.Column</b> parameter to specify whether to shift by rows or columns.
.Column	Specifies whether to shift alternating rows of tiles or columns of tiles. Set to TRUE (-1) to shift columns. Set to FALSE (0) to shift rows which is the default if omitted.
.SeamlessTile	Specifies whether to enable seamless tiling which alters the pattern tiles so that they appear to be continuous. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is TRUE (-1).
.ScaleWithObject	Specifies whether to use absolute and proportional pattern tile sizes. Set to TRUE (-1) to use proportional pattern tile sizes. Set to FALSE (0) to use absolute pattern tile sizes which is the default if omitted. This parameter affects the units used in the <b>.TileWidth</b> and <b>.TileHeight</b> parameters.

#### Note



This command corresponds to clicking Import button in the Two-Color Bitmap Pattern dialog box in Corel VENTURA. Click the



button. This command cannot be used to create a two-color bitmap or use a predefined bitmap as you can with the Two-Color Bitmap Pattern dialog box.



Do not record this command because the first parameter will be filled with the string "Unkown".

#### Example

.GraphicFirst

.FillTwoColor "c:\BIG.bmp", 2, 100, 0, 0, 0, 2, 0, 100, 100, 0, 63500, 63500, 0, 0, 0, FALSE, TRUE, FALSE

The above example fills the first graphic on the active page with the BIG.BMP bitmap, setting the background color to light blue and the foreground color to red using the CMYK color model.

---

{button ,AL('VENTURA\_Fill\_Menu;;;;','0,"Defaultoverview",)} Related Topics

## .FillTwoColorGet (VENTURA)

**.FillTwoColorGet** .FileName=*string*, .BackModel=*long*, .BackColor1=*long*, .BackColor2=*long*, .BackColor3=*long*, .BackColor4=*long*, .FrontModel=*long*, .FrontColor1=*long*, .FrontColor2=*long*, .FrontColor3=*long*, .FrontColor4=*long*, .TileWidth=*long*, .TileHeight=*long*, .TileOffsetX=*long*, .TileOffsetY=*long*, .RowColOffset=*long*, .Column=*Boolean*, .SeamlessTile=*Boolean*, .ScaleWithObject=*Boolean*

This function returns the two-color bitmap pattern fill settings of a selected object. Objects include graphics, text, and frames.

Syntax	Description
.FileName	Omit this parameter since it currently cannot be used to hold a return value.
.BackModel	Specifies the numeric variable that is assigned a return value corresponding to a selected object's background color model: <div><div>1</div><div><a href="#">PANTONE Spot Colors</a></div></div> <div><div>2</div><div><a href="#">CMYK</a></div></div> <div><div>3</div><div><a href="#">CMYK255</a></div></div> <div><div>4</div><div><a href="#">CMY</a></div></div> <div><div>5</div><div>RGB (default if omitted)</div></div> <div><div>6</div><div><a href="#">HSB</a></div></div> <div><div>7</div><div><a href="#">HLS</a></div></div> <div><div>8</div><div><a href="#">Black and White</a></div></div> <div><div>9</div><div><a href="#">Grayscale</a></div></div> <div><div>11</div><div><a href="#">YIQ</a></div></div> <div><div>12</div><div><a href="#">L*a*b*</a></div></div> <div><div>14</div><div><a href="#">PANTONE Hexachrome</a></div></div>

	the width of the bitmap pattern.
.TileOffsetY	Specifies the numeric variable that is assigned the vertical position of the first pattern tile relative to the top of the selected object. This variable is expressed a percentage of the height of the bitmap pattern.
.RowColOffset	Specifies the numeric variable that is assigned the amount to shift the alternating rows or columns of tiles as percentage of the bitmap pattern.
.Column	Specifies the numeric variable that is assigned a value corresponding to whether rows of tiles or columns of tiles are shifted: TRUE (-1) for columns; FALSE (0) for rows.
.SeamlessTile	Specifies the numeric variable that is assigned a value corresponding to whether seamless tiling is enabled: TRUE (-1) if enabled; otherwise, FALSE (0).
.ScaleWithObject	Specifies the numeric variable that is assigned a value corresponding to whether absolute and proportional pattern tile sizes are used: TRUE (-1) for proportional pattern tile sizes; FALSE (0) for absolute pattern tile sizes.

#### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



If more than one object is selected, this function returns the attributes of the first selected object.



If the selected object is not filled with a two-color bitmap pattern fill (for example, a fountain fill), 0 is assigned to each numeric variable used in the function.



This command cannot be recorded.

#### Example

.GraphicFirst

.FillTwoColorGet , BCM&, BC1&, BC2&, BC3&, BC4&, FCM&, FC1&, FC2&, FC3&, FC4&, TW&, TH&, TO&

The above example selects the first graphic on the active page and returns its properties to the specified variables.

---

{button ,AL("VENTURA\_Fill\_Menu;;;;";0,"Defaultoverview"),} Related Topics



**.FillUniform (VENTURA)**

**.FillUniform** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This command sets the uniform fill color of a selected object. Objects include graphics, text, and frames.

Syntax	Description
.ColorModel	<p>Specifies the color model to use for the uniform fill for the selected object:</p> <div><div>1</div><div>PANTONE Spot Colors</div></div> <div><div>2</div><div>CMYK</div></div> <div><div>3</div><div>CMYK255</div></div> <div><div>4</div><div>CMY</div></div> <div><div>5</div><div>RGB (default if omitted)</div></div> <div><div>6</div><div>HSB</div></div> <div><div>7</div><div>HLS</div></div> <div><div>8</div><div>Black and White</div></div> <div><div>9</div><div>Grayscale</div></div> <div><div>11</div><div>YIQ</div></div> <div><div>12</div><div>L*a*b*</div></div> <div><div>14</div><div>PANTONE Hexachrome</div></div>

**Note**

 This command corresponds to the Uniform Fill dialog box in Corel VENTURA. Click the  button.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.FillUniform 2, 40, 0, 20, 60
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame. The ellipse is then filled with a dark green color using the CMYK color model.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.



## .FillUniformGet (VENTURA)

**.FillUniformGet** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This function returns the uniform fill color of a selected object. Objects include graphics, text, and frames.

Syntax	Description
.ColorModel	Specifies the numeric variable that is assigned a return value corresponding to a selected object's uniform color model: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
.Color1	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color3	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color4	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



If more than one object is selected, this function returns the attributes of the first selected object.



If the selected object is not filled with a uniform color (for example, a fountain fill), 0 is assigned to each variable used in the function.



This command cannot be recorded.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FillUniformGet cm&, c1&, c2&, c3&, c4&
```

The above example draws a rectangle and an ellipse. The rectangle is then selected using the **.SelectObjectAt** command. The **.FillUniformGet** function returns the uniform color values for the rectangle to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

The **.LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Fill\_Menu;;;;','0','Defaultoverview',)} [Related Topics](#)

## .GetFillType (VENTURA)

ReturnValue& = .GetFillType ( )

This function returns the fill type for a selected object. Objects include graphics, text, and frames.

Syntax	Description
ReturnValue&	Specifies the numeric variable that is assigned a value corresponding to the fill of the selected object. 0 None 1 Uniform 2 Fountain 6 PostScript 7 Two-color bitmap 8 Monochrome 9 Full Color 10 Vector 11 Texture

### Note



This command cannot be recorded.



If more than one object is selected, this function returns the attributes of the first selected object.

### Example

```
.GraphicFirst  
Ftype = .GetFillType ( )
```

In the above example, the first line selects the first graphic on the current page. The selected graphic's fill type is assigned to the **Ftype** variable.



---

{button ,AL('VENTURA\_Fill\_Menu;;;','0,"Defaultoverview",)} [Related Topics](#)

## .Outline (VENTURA)

**.Outline** .Width=*long*, .Type=*long*, .EndCaps=*long*, .JoinType=*long*, .Aspect=*long*, .Angle=*long*, .DotDash=*long*, .RightArrow=*long*, .LeftArrow=*long*, .BehindFill=*Boolean*

This command sets the outline attributes for a selected graphic or frame.

Syntax	Description
.Width	Specifies the width of the outline in tenths of a micron. Setting this parameter to 0 removes the outline.
.Type	Specifies the style of the outline: 0 None 1 Solid 2 Dotted or Dashed If this parameter is set to Dotted or Dashed, the <b>.DotDash</b> parameter determines the specific style.
.EndCaps	Specifies the line cap style: 0 Butt 1 Round 2 Square
.JoinType	Specifies the corner style: 0 Miter 1 Round 2 Bevel
.Aspect	Specifies the width of the nib. A square nib has a value of 100; smaller values make the nib narrower, creating a calligraphy effect. Valid values range from 1 to 100, inclusive.
.Angle	Specifies the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree). This option creates a calligraphic effect.
.DotDash	Specifies the style of the outline if <b>.Type</b> is set to Dotted or Dashed. The styles correspond to the Style list box in the Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
.RightArrow	Specifies the style of right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.LeftArrow	Specifies the style of left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.BehindFill	Specifies whether the outline is placed behind, or in front of, the object's fill. When placed behind, only half the outline's thickness will be visible. This option is particularly useful for outlined text. Set to TRUE (-1) to enable the behind fill; otherwise, set to FALSE (0), which is the default if omitted.

### Note



This command corresponds to the Outline Pen dialog box in Corel VENTURA. Click the



button.



The **.OutlineColor** command changes an outline's color.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.Outline 0.05*M_INCH, 2, 0, 0, 90, 25, 7
```

The above example draws a rectangle. The rectangle's outline is set to 0.05 inches with a dashed line.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Fill\_Menu;;;;','0,"Defaultoverview",)} [Related Topics](#)



**.OutlineColor (VENTURA)**

**.OutlineColor** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This command sets the outline color for a selected graphic or frame.

Syntax	Description
.ColorModel	<p>Specifies the color model to use for the outline color for the selected object:</p> <div><div>1</div><div>PANTONE Spot Colors</div></div> <div><div>2</div><div>CMYK</div></div> <div><div>3</div><div>CMYK255</div></div> <div><div>4</div><div>CMY</div></div> <div><div>5</div><div>RGB (default if omitted)</div></div> <div><div>6</div><div>HSB</div></div> <div><div>7</div><div>HLS</div></div> <div><div>8</div><div>Black and White</div></div> <div><div>9</div><div>Grayscale</div></div> <div><div>11</div><div>YIQ</div></div> <div><div>12</div><div>L*a*b*</div></div> <div><div>14</div><div>PANTONE Hexachrome</div></div>

**Note**

 This command corresponds to the Outline Color dialog box in Corel VENTURA. Click the  button.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.OutlineColor 2, 40, 0, 20, 60
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame. The ellipse's outline is set to a dark green color using the CMYK color model.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.





# .OutlineColorGet (VENTURA)

**.OutlineColorGet** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This function returns the outline color of a selected graphic or frame.

Syntax	Description
.ColorModel	Specifies the numeric variable that is assigned a return value corresponding to a selected object's outline color model: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
.Color1	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color3	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color4	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  If more than one object is selected, this function returns the attributes of the first selected object.
-  If the selected object does not have an outline, 0 is assigned to each variable used in the function.
-  This command cannot be recorded.

## Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.OutlineColorGet cm&, c1&, c2&, c3&, c4&
```

The above example draws a rectangle and an ellipse. The rectangle is then selected using the **.SelectObjectAt** command. The **.OutlineColorGet** function then returns the outline color values for the rectangle to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.



The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

{button ,AL('VENTURA\_Fill\_Menu;;;;','0,'Defaultoverview',)} [Related Topics](#)





# .OutlineGet (VENTURA)

**.OutlineGet** .Width=*long*, .Type=*long*, .EndCaps=*long*, .JoinType=*long*, .Aspect=*long*, .Angle=*long*, .DotDash=*long*, .RightArrow=*long*, .LeftArrow=*long*, .BehindFill=*Boolean*

This function returns the outline attributes of a selected graphic or frame.

Syntax	Description
.Width	Specifies the numeric variable that is assigned the width of the outline in tenths of a micron.
.Type	Specifies the numeric variable that is assigned the style of the outline: 0     None 1     Solid 2     Dotted or Dashed  If this parameter assigns Dotted or Dashed, the <b>.DotDash</b> parameter assigns the specific style.
.EndCaps	Specifies the numeric variable that is assigned the line cap style: 0     Butt 1     Round 2     Square
.JoinType	Specifies the numeric variable that is assigned the corner style: 0     Miter 1     Round 2     Bevel
.Aspect	Specifies the numeric variable that is assigned the width of the nib.
.Angle	Specifies the numeric variable that is assigned the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree).
.DotDash	Specifies the numeric variable that is assigned the style of the outline if <b>.Type</b> is set to Dotted or Dashed. The styles correspond to the Style list box in the Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
.RightArrow	Specifies the numeric variable that is assigned the style of right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.LeftArrow	Specifies the numeric variable that is assigned the style of left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.BehindFill	Specifies the numeric variable that is assigned a value indicating whether the behind fill option is enabled: TRUE (-1) if enabled; otherwise, FALSE (0).

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  If more than one object is selected, this function returns the attributes of the first selected object.
-  If the selected object does not have an outline, 0 is assigned to each variable used in the function.
-  This command cannot be recorded.

## Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.OutlineGet Owidth&, Otype&, , , , Odotdash&
```

The above example draws a rectangle and an ellipse. The rectangle is then selected using the **.SelectObjectAt** command. The **.OutlineGet** function then returns the outline attributes for the rectangle to the variables **Owidth**, **Otype**, and **Odotdash**.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

**.OutlineOverprint (VENTURA)**

**.OutlineOverprint .Overprint=*Boolean*,**

This command sets the overprint option for a selected object's outline. Objects include graphics and frames. Enabling the overprint option allows the outline to print over the background color. This option is relevant only when printing color separations.

Syntax	Description
<b>.Overprint</b>	Specifies whether to enable the overprint option. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).

**Example**  
M\_INCH = LENGTHCONVERT (1 , 7 , 1)  
.DrawRectangle 2\*M\_INCH, 2\*M\_INCH, 2.5\*M\_INCH, 4.3\*M\_INCH  
.Outline 0.05\*M\_INCH, 2, 0, 0, 90, 25, 7  
.OutlineOverprint TRUE

The above example draws a rectangle. The rectangle's outline is set to 0.05 inches with a dashed line. The outline then has the overprint option applied to it.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

**.OutlineOverprintGet (VENTURA)**

ReturnValue = .OutlineOverprintGet ( )

This function returns the overprint outline setting of a selected object. Objects include graphics, text, and frames.

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a value corresponding to whether the overprint option is enabled: TRUE (-1) if enabled; otherwise FALSE (0).

**Note**

- 1**
- If more than one object is selected, this function returns the attributes of the first selected object.
- 1**
- This command cannot be recorded.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.OutlineGet Owidth, Otype, , , , Odotdash
OPoption = .OutlineOverprintGet ( )
```

The above example draws a rectangle and an ellipse. The rectangle is then selected using the .SelectObjectAt command. The .OutlineGet function returns the outline attributes for the rectangle to the variables **Owidth**, **Otype**, and **Odotdash**. The .OutlineOverprintGet function returns the overprint status to the **OPoption** variable.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.



## .EditFindTag (VENTURA)

**Command:** .EditFindTag

**Function:** ReturnValue = .EditFindTag ( )

This command and function searches for the next occurrence of a paragraph or character tag specified in a previously executed Find & Replace initialization command ([.EditInitFindTag](#)).

The **.EditFindTag** command or function corresponds to clicking the "Find Next" button in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace), while the **.EditInitFindTag** corresponds to setting options in the Find & Replace dialog box.

Syntax	Description
<b>ReturnValue</b>	Specifies the numeric variable that is assigned a value corresponding to whether the tag specified in the <a href="#">.EditInitFindTag</a> command was found: TRUE (-1) if found; otherwise FALSE (0).

### Note



The search options (for example, direction) for this command are set in a preceding [.EditInitFindTag](#) command.

### Example

```
.EditInitFindTag "MainHeading"  
.EditFindTag
```

The above example searches for the next occurrence of the MainHeading paragraph tag in the current document.

```
.EditInitFindTag "PartNumber", , -1, -1, -1  
.EditFindTag
```

The above example searches for the next occurrence of PartNumber character tag in the current chapter. The search is performed backwards.

```
.EditInitFindTag "MainHeading", "MinorHeading"  
.EditFindTag  
.EditReplace 0
```

The above example searches for the next occurrence of the MainHeading paragraph tag in the current document and replaces it with the MinorHeading paragraph tag.

```
' example using EditFindTag as a function  
.EditInitFindTag "PartNumber", , -1, -1  
DO  
    Found = .EditFindTag()  
    i = i + 0  
LOOP UNTIL Found = FALSE  
MESSAGE "Number of times PartNumber tag found: " & i
```

The above example counts the number of times the character tag PartNumber is found in the current chapter. The **DO...LOOP** is exited once the **.EditFindTag** function returns FALSE.

---

{button ,AL('VENTURA\_EditFindTag\_Menu;vent\_find;;;',0,"Defaultoverview"),} [Related Topics](#)

## .EditFindText (VENTURA)

**Command:** .EditFindText


**Function:** ReturnValue = .EditFindText ( )

This command and function searches for the next occurrence of text, text attributes, or both, specified in a previously executed Find & Replace initialization command. Core! SCRIPT for VENTURA uses three initialization commands: .EditInitFindText, .EditInitFindAttribute, and .EditInitFindColor. These three commands correspond to setting options in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace).

The **.EditFindText** command or function corresponds to clicking the "Find Next" button in the Find & Replace dialog box.

Syntax	Description
<b>ReturnValue</b>	Specifies the numeric variable that is assigned a value corresponding to whether the text specified in the <u>.EditInitFindText</u> command was found: TRUE (-1) if found; otherwise FALSE (0).

### Note

 The search and attribute options (for example, direction and match case options) for this command are set in a preceding .EditInitFindText command.

### Example

```
.EditInitFindText "dog"  
.EditFindText
```

The above example searches for the next occurrence of **dog** in the text.

```
.EditInitFindText "Dog", , -1, -1,  
.EditFindText
```

The above example searches for the next occurrence of **Dog** in the text. Only the text in the current chapter is searched, and the search is performed backwards.

```
.EditInitFindText "Dog", "Cat", -1, -1,  
.EditFindText  
.EditReplace 0
```

The above example is the same as the previous example, except that the first occurrence of **Dog** is replaced with **Cat**.

```
' example using EditFindText as a function  
.EditInitFindText "Dog", , -1, , , -1, -1  
DO  
    Found = .EditFindText()  
    i = i + 0  
LOOP UNTIL Found = FALSE  
MESSAGE "Number of times Dog found: " & i
```

The above example counts the number of times the word **Dog** is found in the current chapter. The DO...LOOP is exited once the **.EditFindText** function returns FALSE.

---

{button ,AL('VENTURA\_EditFindText\_Menu;vent\_find;;;','0,"Defaultoverview",)} Related Topics

## .EditInitFindAttribute (VENTURA)

**.EditInitFindAttribute** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .Kerning=*long*, .ShiftUp=*long*, .Underline=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*, .Uppercase=*Boolean*, .Effect=*long*

This command sets Find text attribute parameters for subsequent **.EditReplace** and **.EditFindText** commands.

This command corresponds to setting the text attributes in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace). The **.EditFindText** and **.EditReplace** commands correspond to clicking the "Find Next", "Replace", and "Replace All" buttons in the Find & Replace dialog box.

Syntax	Description																		
.FontName	Specifies the name of the font to find.																		
.PointSize	Specifies the font size in points to find.																		
.Weight	<p>Specifies the font weight setting (number of inked pixels per 1000 pixels) to find. The weight setting ranges from 0 to 1000. Common values and their corresponding names include:</p> <table><tr><td>100</td><td>Thin</td></tr><tr><td>200</td><td>Extra Light, Ultra Light</td></tr><tr><td>300</td><td>Light</td></tr><tr><td>400</td><td>Normal, Regular</td></tr><tr><td>500</td><td>Medium</td></tr><tr><td>600</td><td>Semi Bold, Demi Bold</td></tr><tr><td>700</td><td>Bold</td></tr><tr><td>800</td><td>Extra Bold, Ultra Bold</td></tr><tr><td>900</td><td>Black, Heavy</td></tr></table> <p>Some fonts may use different names than those shown above for the same weight settings.</p>	100	Thin	200	Extra Light, Ultra Light	300	Light	400	Normal, Regular	500	Medium	600	Semi Bold, Demi Bold	700	Bold	800	Extra Bold, Ultra Bold	900	Black, Heavy
100	Thin																		
200	Extra Light, Ultra Light																		
300	Light																		
400	Normal, Regular																		
500	Medium																		
600	Semi Bold, Demi Bold																		
700	Bold																		
800	Extra Bold, Ultra Bold																		
900	Black, Heavy																		
.Italic	Specifies to search for italic formatting. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is FALSE.																		
.Kerning	Specifies to search for a specified amount of kerning in hundredths of an em. An em is a unit of measure roughly equal to the width of the capital M at the current font size.																		
.ShiftUp	Specifies to search for a distance text is moved off baseline. This value is expressed in inches multiplied by 300, rounded to the nearest whole number. To express this value using points, multiply the number of points by 300, divide by 72, and round to the nearest whole number.																		
.Underline	<p>Specifies to search for text using underline attributes:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Single</td></tr><tr><td>2</td><td>Double</td></tr><tr><td>3</td><td>Word underline</td></tr></table>	0	None	1	Single	2	Double	3	Word underline										
0	None																		
1	Single																		
2	Double																		
3	Word underline																		
.StrikeThru	Specifies to search for strike-thru formatting. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is FALSE.																		
.Overscore	Specifies to search for overscore formatting. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is FALSE.																		
.Uppercase	Specifies to search for uppercase formatting. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is FALSE.																		
.Effect	<p>Specifies to search for text using effect attributes:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Superscript</td></tr><tr><td>2</td><td>Subscript</td></tr><tr><td>3</td><td>Small caps</td></tr></table>	0	None	1	Superscript	2	Subscript	3	Small caps										
0	None																		
1	Superscript																		
2	Subscript																		
3	Small caps																		

### Note

**1** If you are using multiple **.EditInitFindAttribute** commands in a script (for example, in a loop), you should precede the **.EditInitFindAttribute** commands with the following script statement to reset the search criteria:  
**.EditInitFindText .Reset=TRUE**

**1** Use the **.EditInitReplaceAttribute** command to set Replace text attribute parameters for subsequent **.EditReplace** and **.EditFindText** commands.

### Example

```
.EditInitFindAttribute "Courier", 10, 400, -1  
.EditFindText
```

The above example searches for the next occurrence of a normal-italic Courier font at 10 points in the text.

```
.EditInitFindAttribute "Courier", 10, 400, -1  
.EditInitFindText , , , -1  
.EditFindText
```

The above example is the same as the previous example except the font is searched for backwards by using the **.EditInitFindText** command.

```
.EditInitFindAttribute "Courier", 10, 400, -1  
.EditInitReplaceAttribute "Arial", 12, 700  
.EditReplace -1
```

The above example searches for all occurrences of normal-italic Courier font at 10 points in the text and replaces it with bold Arial font at 12 points.

```
.EditInitFindAttribute "Courier", 10, 400, -1  
.EditInitReplaceAttribute "Arial", 12, 700  
.EditInitFindText "dog", "cat",  
.EditReplace -1
```

The above example searches for all occurrences of **dog** in a normal-italic Courier font at 10 points in the text and replaces it with **cat** in a bold Arial font at 12 points.

---


{button ,AL('VENTURA\_EditInitFindAttribute\_Menu;vent\_find;;;','0,"Defaultoverview",')} [Related Topics](#)

## .EditInitFindColor (VENTURA)


**.EditInitFindColor** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*


This command sets Find text color parameters for subsequent **.EditReplace** and **.EditFindText** commands.

This command corresponds to setting the text color attributes in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace). The **.EditFindText** and **.EditReplace** commands correspond to clicking the "Find Next", "Replace", and "Replace All" buttons in the Find & Replace dialog box.

Syntax	Description
.ColorModel	<p>Specifies the color model to use for the text to find:</p> <ol style="list-style-type: none"><li>1 <u>PANTONE Spot Colors</u></li><li>2 <u>CMYK</u></li><li>3 <u>CMYK255</u></li><li>4 <u>CMY</u></li><li>5 <u>RGB</u> (default if omitted)</li><li>6 <u>HSB</u></li><li>7 <u>HLS</u></li><li>8 <u>Black and White</u></li><li>9 <u>Grayscale</u></li><li>11 <u>YIQ</u></li><li>12 <u>L*a*b*</u></li><li>14 <u>PANTONE Hexachrome</u></li></ol> <p>If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.</p>
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### Note

 If you are using multiple **.EditInitFindColor** commands in a script (for example, in a loop), you should precede the **.EditInitFindColor** commands with the following script statement to reset the search criteria:  
**.EditInitFindText .Reset=TRUE**

 Use the **.EditInitReplaceColor** command to set Replace text color parameters for subsequent **.EditReplace** and **.EditFindText** commands.

### Example

```
.EditInitFindColor 5, 200, 10, 15
.EditFindText
```

The above example searches for the next occurrence of a specified red colored font that uses the RGB color model.

```
.EditInitFindColor 5, 200, 10, 15
.EditInitFindText , , , -1
.EditFindText
```

The above example is the same as the previous example except the font is searched for backwards by using the **.EditInitFindText** command.

```
.EditInitFindColor 5, 200, 10, 15
.EditInitReplaceColor 5, 22, 22, 255
.EditReplace -1
```

The above example searches for all occurrences of a specified red colored font that uses the RGB color model and replaces it with a blue font that also uses the RGB color model.

```
.EditInitFindColor 5, 200, 10, 15  
.EditInitReplaceColor 5, 22, 22, 255  
.EditInitFindText "dog", "cat",  
.EditReplace -1
```

The above example searches for all occurrences of **dog** in a red font and replaces it with **cat** in a blue font.

---

{button ,AL('VENTURA\_EditInitFindAttribute\_Menu;vent\_find;;;',0,"Defaultoverview"),} [Related Topics](#)

## .EditInitFindTag (VENTURA)

**.EditInitFindTag** .FindTag=*string*, .ReplaceTag=*string*, .CharacterTag=*Boolean*, .CurrentChapter=*Boolean*, .Backward=*Boolean*, .FromBegin=*Boolean*


This command sets Find and Find & Replace tag parameters for subsequent **.EditReplace** and **.EditFindTag** commands.

This command corresponds to setting the tag attributes in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace). The **.EditFindTag** and **.EditReplace** commands correspond to clicking the "Find Next", "Replace", and "Replace All" buttons in the Find & Replace dialog box.

For more options, see the **.EditInitFindTagOptions** command which was introduced in Corel VENTURA 8.

Syntax	Description
.FindTag	Specifies the paragraph or character tag to search for.
.ReplaceTag	Optional parameter that specifies the replacement paragraph or character tag. This tag must exist in the document this command is being performed in.
.CharacterTag	Specifies whether to search for a character tag or a paragraph tag. Set to TRUE (-1) to search for a character tag; set to FALSE (0) to search for a paragraph tag (default if omitted).
.CurrentChapter	Specifies whether to search the current chapter or all the chapters in a multi-chapter document. Set to TRUE (-1) to search the current chapter. Set to FALSE (0) to search all chapters (default if omitted).
.Backward	Specifies the search direction. Set to TRUE (-1) to search backward. Set to FALSE (0) to search forward (default if omitted).
.FromBegin	Specifies whether to search from the beginning of the document or current chapter (if <b>.CurrentChapter</b> is TRUE). Set to TRUE (-1) to search from the beginning. Set to FALSE (0) to search from current position (default if omitted).

### Note

 If you are using multiple **.EditInitFindTag** commands in a script (for example, in a loop), you should precede the **.EditInitFindTag** commands with the following script statement to reset the search criteria:  
`.EditInitFindText .Reset=TRUE`

### Example

```
.EditInitFindTag "MainHeading"  
.EditFindTag
```

The above example searches for the next occurrence of the **MainHeading** paragraph tag in the current document.

```
.EditInitFindTag "PartNumber", , -1, -1, -1  
.EditFindTag
```

The above example searches for the next occurrence of **PartNumber** character tag in the current chapter. The search is performed backwards.

```
.EditInitFindTag "MainHeading", "MinorHeading"  
.EditFindTag  
.EditReplace 0
```

The above example searches for the next occurrence of the **MainHeading** paragraph tag in the current document and replaces it with the MinorHeading paragraph tag.

```
.EditInitFindTag "MainHeading", "MinorHeading", , -1  
.EditFindTag  
.EditReplace -1
```

The above example is the same as the previous example but all occurrences of the **MainHeading** paragraph tag are replaced in the current chapter.

---

{button ,AL('VENTURA\_EditInitFindTag\_Menu;vent\_find;;;',0,"Defaultoverview",)} [Related Topics](#)

# .EditInitFindTagOptions (VENTURA)



**.EditInitFindTagOptions** .FindTag=*string*, .ReplaceTag=*string*, .TagType=*long*, .CurrentChapter=*Boolean*, .Backward=*Boolean*, .FromBegin=*Boolean*

This command sets Find and Find & Replace tag parameters for subsequent **.EditReplace** and **.EditFindTag** commands.

This command corresponds to setting the tag attributes in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace). The **.EditFindTag** and **.EditReplace** commands correspond to clicking the "Find Next", "Replace", and "Replace All" buttons in the Find & Replace dialog box.

Syntax	Description
.FindTag	Specifies the paragraph or character tag to search for.
.ReplaceTag	Optional parameter that specifies the replacement paragraph or character tag. This tag must exist in the document this command is being performed in.
.TagType	Specifies the tag type to search for: 0 paragraph tag 1 character tag 2 frame tag
.CurrentChapter	Specifies whether to search the current chapter or all the chapters in a multi-chapter document. Set to TRUE (-1) to search the current chapter. Set to FALSE (0) to search all chapters (default if omitted).
.Backward	Specifies the search direction. Set to TRUE (-1) to search backward. Set to FALSE (0) to search forward (default if omitted).
.FromBegin	Specifies whether to search from the beginning of the document or current chapter (if <b>.CurrentChapter</b> is TRUE). Set to TRUE (-1) to search from the beginning. Set to FALSE (0) to search from current position (default if omitted).

## Note

-  This command was introduced in Corel VENTURA 8.
-  If you are using multiple **.EditInitFindTag** commands in a script (for example, in a loop), you should precede the **.EditInitFindTag** commands with the following script statement to reset the search criteria:  
`.EditInitFindText .Reset=TRUE`

## Example

```
.EditInitFindTagOptions "MainHeading"  
.EditFindTag
```

The above example searches for the next occurrence of the **MainHeading** paragraph tag in the current document.

```
.EditInitFindTagOptions "PartNumber", , 1, -1, -1  
.EditFindTag
```

The above example searches for the next occurrence of **PartNumber** character tag in the current chapter. The search is performed backwards.

```
.EditInitFindTagOptions "MainHeading", "MinorHeading"  
.EditFindTag  
.EditReplace 0
```

The above example searches for the next occurrence of the **MainHeading** paragraph tag in the current document and replaces it with the MinorHeading paragraph tag.

```
.EditInitFindTagOptions "MainHeading", "MinorHeading", 2, -1  
.EditFindTag  
.EditReplace -1
```

The above example is the same as the previous example but all occurrences of the **MainHeading** paragraph tag are replaced in the current chapter.





## .EditInitFindText (VENTURA)


**.EditInitFindText** *.FindText=string*, *.EditReplace=string*, *.CurrentChapter=Boolean*, *.Backward=Boolean*, *.FromBegin=Boolean*, *.MatchCase=Boolean*, *.WholeWord=Boolean*, *.MaintainCase=Boolean*, *.Reset=Boolean*

This command sets Find and Find & Replace text parameters for subsequent **.EditReplace** and **.EditFindText** commands.


This command corresponds to setting the text color attributes in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace). The **.EditFindText** and **.EditReplace** commands correspond to clicking the "Find Next", "Replace", and "Replace All" buttons in the Find & Replace dialog box.

Syntax	Description
<b>.FindText</b>	<p>Specifies the text to search for. Use wildcards to search for text with letters in common. To find any single character, use ?. For example, b?t finds <b>bat</b> and <b>bet</b>. To find any string of characters, use *. For example, b*d finds <b>bad</b> and <b>battered</b>.</p> <p>To search for special characters, click  for the corresponding special codes. For example, to find a Paragraph Return, use &lt;P&gt;.</p> <p>To search for special items, click  for the corresponding special codes. For example, to an Index Entry, use &lt;I&gt;.</p>
<b>.ReplaceText</b>	<p>Optional parameter that specifies the replacement text for the found text. To delete the text specified in the <b>.FindText</b> parameter, set this parameter to an empty string ("").</p> <p>See the <b>.FindText</b> parameter for information about special characters and items.</p>
<b>.CurrentChapter</b>	<p>Specifies whether to search the current chapter or all the chapters in a multi-chapter document. Set to TRUE (-1) to search the current chapter. Set to FALSE (0) to search all chapters (default if omitted).</p>
<b>.Backward</b>	<p>Specifies the search direction. Set to TRUE (-1) to search backward. Set to FALSE (0) to search forward (default if omitted).</p>
<b>.FromBegin</b>	<p>Specifies whether to search from the beginning of the document or current chapter (if <b>.CurrentChapter</b> is TRUE). Set to TRUE (-1) to search from the beginning. Set to FALSE (0) to search from current position (default if omitted).</p>
<b>.MatchCase</b>	<p>Specifies whether to search for text with the same capitalization as the text specified in the <b>.FindText</b> parameter. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted.</p>
<b>.WholeWord</b>	<p>Specifies whether to match only whole words with the text specified in the <b>.FindText</b> parameter. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted.</p>
<b>.MaintainCase</b>	<p>Specifies whether to give the replacement text the same capitalization as the text specified in the <b>.FindText</b> parameter. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted.</p>
<b>.Reset</b>	<p>If this command is used more than once in a script, this parameter specifies whether to reset all parameters or retain the parameter values for non-specified parameters. Set to TRUE (-1) to reset the parameter settings in this command (default if omitted). Set to FALSE (0) to retain the parameter settings in this command. See the last example in the list below for more information about using this command.</p>

### Note

 After using this command to find or replace text, you should insert the following command at the end of your script (this will reset the Find & Replace parameters when running your script again or any other scripts):

```
.EditInitFindText .Reset=FALSE
```

 This command also sets the Find & Replace search and attributes options for the **.EditInitFindAttribute**, **.EditInitReplaceAttribute**, **.EditInitFindColor**, and **.EditInitReplaceColor** commands.

### Example

```
.EditInitFindText "dog"  
.EditFindText
```

The above example searches for the next occurrence of **dog** in the text.

```
.EditInitFindText "Dog", , -1, -1,  
.EditFindText
```

The above example searches for the next occurrence of **Dog** in the text. Only the text in the current chapter is searched, and the search is performed backwards.

```
.EditInitFindText "Dog", "Cat", -1, -1,
.EditFindText
.EditReplace 0
```

The above example is the same as the previous example but the first occurrence of **Dog** is replaced with **Cat**.

```
.EditInitFindText "Dog", "Cat", -1, -1,
.EditFindText
.EditReplace -1
```

The above example replaces all occurrences of **Dog** with **Cat**, backwards from the position of the insertion point in the current chapter.

```
'Example for the .Reset parameter
.EditInitFindText "dog", "Cat", -1, -1, 0, -1, 0, -1 'first EditInitFindText command
'block of other script statements
.EditInitFindText , "Cat", , , , , , FALSE 'second EditInitFindText command
```

In the above example, the second line initializes the find text parameters. The third line indicates a block of scripting commands. The third line changes the **.EditReplace** parameter to "Cat" but retains all other previous parameter settings because the **.Reset** parameter is set to FALSE.

---

{button ,AL('VENTURA\_EditInitFindText\_Menu;vent\_find;;;','0,"Defaultoverview",')} [Related Topics](#)

## .EditInitReplaceAttribute (VENTURA)

**.EditInitReplaceAttribute** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .Kerning=*long*, .ShiftUp=*long*, .Underline=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*, .Uppercase=*Boolean*, .Effect=*long*

This command sets Replace text attribute parameters for a subsequent **.EditReplace** commands.

This command corresponds to setting the text attributes in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace). The **.EditReplace** commands correspond to clicking the "Replace" and "Replace All" buttons in the Find & Replace dialog box.

Syntax	Description																		
.FontName	Specifies the font to apply to the found text. If omitted, the found text's font is not changed.																		
.PointSize	Specifies the font size in points to apply to the found text. If omitted, the found text's point size is not changed.																		
.Weight	<p>Specifies the font weight setting (number of inked pixels per 1000 pixels) to apply to the found text. If omitted, the found text's weight setting is not changed.</p> <p>The weight setting ranges from 0 to 1000. Common values and their corresponding names include:</p> <table><tr><td>100</td><td>Thin</td></tr><tr><td>200</td><td>Extra Light, Ultra Light</td></tr><tr><td>300</td><td>Light</td></tr><tr><td>400</td><td>Normal, Regular</td></tr><tr><td>500</td><td>Medium</td></tr><tr><td>600</td><td>Semi Bold, Demi Bold</td></tr><tr><td>700</td><td>Bold</td></tr><tr><td>800</td><td>Extra Bold, Ultra Bold</td></tr><tr><td>900</td><td>Black, Heavy</td></tr></table> <p>Some fonts may use different names than those shown above for the same weight settings.</p> <p>This parameter is ignored if the font you're applying this setting to doesn't have the specified weight installed on your system. For example, Courier is normally available with only Normal (400) and Bold (700) weight settings. If you set the weight for a Courier font to a value other than 400 or 700, this parameter is ignored and the default setting is used.</p>	100	Thin	200	Extra Light, Ultra Light	300	Light	400	Normal, Regular	500	Medium	600	Semi Bold, Demi Bold	700	Bold	800	Extra Bold, Ultra Bold	900	Black, Heavy
100	Thin																		
200	Extra Light, Ultra Light																		
300	Light																		
400	Normal, Regular																		
500	Medium																		
600	Semi Bold, Demi Bold																		
700	Bold																		
800	Extra Bold, Ultra Bold																		
900	Black, Heavy																		
.Italic	Specifies the italic setting to apply to the found text. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the found text's italic setting is not changed.																		
.Kerning	Specifies the amount of kerning to apply to the found text in hundredths of an em. An em is a unit of measure roughly equal to the width of the capital M at the current font size. Since the value is relative to the font size, you do not need to change the setting when the font size changes. If omitted, the found text's kerning setting is not changed.																		
.ShiftUp	Specifies the distance to move the found text off the baseline. This value is expressed in inches multiplied by 300, rounded to the nearest whole number. To express this value using points, multiply the number of points by 300, divide by 72, and round to the nearest whole number.																		
.Underline	<p>Specifies the underline setting to apply to the found text:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Superscript</td></tr><tr><td>2</td><td>Subscript</td></tr><tr><td>3</td><td>Small caps</td></tr></table> <p>If omitted, the found text's underline setting is not changed.</p>	0	None	1	Superscript	2	Subscript	3	Small caps										
0	None																		
1	Superscript																		
2	Subscript																		
3	Small caps																		
.StrikeThru	Specifies the strike-thru setting to apply to the found text. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the found text's strike-thru setting is not changed.																		
.Overscore	Specifies the overscore setting to apply to the found text. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the found text's overscore setting is not changed.																		
.Uppercase	Specifies the uppercase formatting setting to apply to the found text. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the found text's uppercase formatting setting is not changed.																		
.Effect	<p>Specifies the effect setting to apply to the found text:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Superscript</td></tr></table>	0	None	1	Superscript														
0	None																		
1	Superscript																		

- 2 Subscript
- 3 Small caps

If omitted, the found text's effect setting is not changed.

#### Note



Use the **EditInitFindAttribute** command to set Find text attribute parameters for subsequent **EditReplace** and **EditFindText** commands.

#### Example

```
.EditInitFindAttribute "Courier", 10, 400, -1
.EditFindText
```

The above example searches for the next occurrence of a normal-italic Courier font at 10 points in the text.

```
.EditInitFindAttribute "Courier", 10, 400, -1
.EditInitFindText , , -1
.EditFindText
```

The above example is the same as the previous example except the font is searched for backwards by using the **EditInitFindText** command.

```
.EditInitFindAttribute "Courier", 10, 400, -1
.EditInitReplaceAttribute "Arial", 12, 700
.EditReplace -1
```

The above example searches for all occurrences of normal-italic Courier font at 10 points in the text and replaces it with bold Arial font at 12 points.

```
.EditInitFindAttribute "Courier", 10, 400, -1
.EditInitReplaceAttribute "Arial", 12, 700
.EditInitFindText "dog", "cat",
.EditReplace -1
```

The above example searches for all occurrences of **dog** in a normal-italic Courier font at 10 points in the text and replaces it with **cat** in a bold Arial font at 12 points.

---

{button ,AL("VENTURA\_EditInitReplaceAttribute\_Menu;vent\_find;;;",0,"Defaultoverview"),} [Related Topics](#)

## **.EditInitReplaceColor (VENTURA)**

**.EditInitReplaceAttribute** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This command sets Replace text color attribute parameters for a subsequent **.EditReplace** commands.

This command corresponds to setting the text color attributes in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace). The **.EditReplace** commands correspond to clicking the "Replace" and "Replace All" buttons in the Find & Replace dialog box.

Syntax	Description
.ColorModel	Specifies the color model to apply to the found text.: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u> If omitted, the found text's color model is not changed.
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### **Note**

**I** Use the **.EditInitFindColor** command to set Find text color parameters for subsequent **.EditReplace** and **.EditFindText** commands.

### **Example**

```
.EditInitFindColor 5, 200, 10, 15
.EditFindText
```

The above example searches for the next occurrence of a specified red colored font that uses the RGB color model.

```
.EditInitFindColor 5, 200, 10, 15
.EditInitFindText , , , -1
.EditFindText
```

The above example is the same as the previous example, except that the font is searched for backwards using the **.EditInitFindText** command.

```
.EditInitFindColor 5, 200, 10, 15
.EditInitReplaceColor 5, 22, 22, 255
.EditReplace -1
```

The above example searches for all occurrences of a specified red colored font that uses the RGB color model and replaces it a blue font that also uses the RGB color model.

```
.EditInitFindColor 5, 200, 10, 15
.EditInitReplaceColor 5, 22, 22, 255
.EditInitFindText "dog", "cat",
.EditReplace -1
```

The above example searches for all occurrences of **dog** in a red font and replaces it with **cat** in a blue font.

---

{button ,AL('VENTURA\_EditInitReplaceAttribute\_Menu;vent\_find;;;','0,"Defaultoverview",)} Related Topics

## .EditReplace (VENTURA)

**.EditReplace** .ReplaceAll=*Boolean*

This command replaces text, text attributes, and paragraph and character tags. The **.EditReplace** command corresponds to clicking the "Replace" or "Replace All" buttons in the Find & Replace dialog box in VENTURA (click Edit, Find & Replace).

This command replaces text specified in a previously executed **.EditInitFindText** command with other text specified with the same **.EditInitFindText** command. The search and attribute options (for example, direction and match case options) for this command are also set in the same preceding **.EditInitFindText** command. The **.EditInitFindText** command corresponds to setting the text values in the Find & Replace dialog box.

This command also replaces text attributes specified in a previously executed **.EditInitFindAttribute**, **.EditInitReplaceAttribute**, **.EditInitFindColor**, and **.EditInitReplaceColor** commands. See these commands for more information and examples.

For information and examples of replacing found paragraph or character tags, see the **.EditInitFindTag** command.

Syntax	Description
.ReplaceAll	Specifies whether to the next found occurrence or all occurrences of the found text. Set to TRUE (-1) to replace all found occurrences; set to FALSE (0) to replace the next found occurrence (default if omitted).

### Example

```
.EditInitFindText "dog"
.EditFindText
```

The above example searches for the next occurrence of **dog** in the text.

```
.EditInitFindText "Dog", , -1, -1,
.EditFindText
```

The above example searches for the next occurrence of **Dog** in the text. Only the text in the current chapter is searched, and the search is performed backwards.

```
.EditInitFindText "Dog", "Cat", -1, -1,
.EditFindText
.EditReplace 0
```

The above example is the same as the previous example but the first occurrence of **Dog** is replaced with **Cat**.

```
.EditInitFindText "Dog", "Cat", -1, -1,
.EditFindText
.EditReplace -1
```

The above example replaces all occurrences of **Dog** with **Cat**, backwards from the position of the insertion point in the current chapter.

---

{button ,AL("VENTURA\_EditReplace\_Menu;vent\_find;;;",0,"Defaultoverview"),} [Related Topics](#)

## .FormatCondition (VENTURA)

**.FormatCondition .Condition=string**

This command applies a condition to the active or selected item in the active VENTURA document. Items include frames, chapters, paragraph tags, and tables.

Syntax	Description
<b>.Condition</b>	Specifies the condition that is applied to the active or selected item. Multiple conditions must be separated by logical condition operator symbols: + And   Or ! Not

### Note



This command corresponds to the Object Conditions dialog box in Corel VENTURA. Click Format, Object Conditions.



Use the **.FormatPubCondition** command to create conditions.

### Example

```
.FormatCondition "Con1"
```

The above example applies the **Con1** condition to the active or selected item.

```
.FormatCondition "Con1+Con2"
```

The above example applies the **Con1** and **Con2** conditions to the active or selected item.

---

{button ,AL('VENTURA\_FormatCrossReference\_Menu;vent\_format;;;',0,"Defaultoverview"),} [Related Topics](#)



## .FormatConditionGet (VENTURA)

**.FormatConditionGet .Condition=string**

This function returns the condition applied to the active or selected item in the active VENTURA document. Items include frames, chapters, paragraph tags, and tables.

Syntax	Description
<b>.Condition</b>	Specifies the string variable that is assigned the condition that is applied to the active or selected item. An empty string is assigned if a condition hasn't been applied.

### Note



The variable specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM Active_Con AS STRING
.FormatConditionGet Active_Con
```

The above example assigns the condition that is applied to the active or selected item to the **Active\_Con** variable.

---

{button ,AL('VENTURA\_FormatCrossReference\_Menu;vent\_format;;;',0,"Defaultoverview"),} [Related Topics](#)

## .FormatObjectBegin (VENTURA)

**.FormatObjectBegin .ObjectType=long**

Together, the **FormatObjectBegin** and **FormatObjectEnd** commands are required to modify the properties of a paragraph, frame, or page. The two commands enclose Corel VENTURA property modifying commands. The **FormatObjectBegin** statement is placed before the first modifying command and the **FormatObjectEnd** command is placed after the last modifying command.

To modify the properties of a paragraph tag, frame tag, or page tag, see the following commands: [FormatParaTagBegin](#), [FormatFrameTagBegin](#), and [FormatPageTagBegin](#).

Syntax	Description
<b>.ObjectType</b>	Specifies the item type to modify: 1 Paragraph 2 Frame 3 Page 4 Current selection

### Note



This command was introduced in Corel VENTURA 8.



For a listing of paragraph modifying commands, see [FormatParaTagBegin](#).



For a listing of frame modifying commands, see [FormatFrameTagBegin](#).



For a listing of paragraph modifying commands, see [FormatPageTagBegin](#).

### Example

```
.FormatObjectBegin 1
.FormatParaColor -1, 5, 0, 30, 240
.FormatParaHyphenation -1, "Italian", , 4
.FormatObjectEnd
```

The above example shows how to modify the properties of a paragraph.

```
.FormatObjectBegin 2
.FormatFrameVerticalRule 3, 5, -1
.FormatFrameTypography 2, 3, , , , 1, 0.25*M_INCH, 0.25*M_INCH
.FormatObjectEnd
```

The above example shows how to modify the properties of a frame.

```
.FormatPageTagBegin "Sample"
.FormatPageLayout 34, TRUE, 5*M_INCH, 7*M_INCH
.FormatPageTagEnd
```

The above example shows how to modify the properties of a page.

---

{button ,AL("VENTURA\_FormatCrossReference\_Menu;vent\_format;;;0,"Defaultoverview"),} [Related Topics](#)

## **.FormatObjectEnd (VENTURA)**

### **.FormatObjectEnd**

Together, the **FormatObjectBegin** and **FormatObjectEnd** commands are required to modify the properties of a paragraph, frame, or page. The two commands enclose Corel VENTURA property modifying commands. The **FormatObjectBegin** statement is placed before the first modifying command and the **FormatObjectEnd** command is placed after the last modifying command.

To modify the properties of a paragraph tag, frame tag, or page tag, see the following commands: **FormatParaTagBegin**, **FormatFrameTagBegin**, and **FormatPageTagBegin**.

#### **Note**



This command was introduced in Corel VENTURA 8.



For a listing of paragraph modifying commands, see **FormatParaTagBegin**.



For a listing of frame modifying commands, see **FormatFrameTagBegin**.



For a listing of paragraph modifying commands, see **FormatPageTagBegin**.

#### **Example**

```
.FormatObjectBegin 1
.FormatParaColor -1, 5, 0, 30, 240
.FormatParaHyphenation -1, "Italian", , 4
.FormatObjectEnd
```

The above example shows how to modify the properties of a paragraph.

```
.FormatObjectBegin 2
.FormatFrameVerticalRule 3, 5, -1
.FormatFrameTypography 2, 3, , , , 1, 0.25*M_INCH, 0.25*M_INCH
.FormatObjectEnd
```

The above example shows how to modify the properties of a frame.

```
.FormatPageTagBegin "Sample"
.FormatPageLayout 34, TRUE, 5*M_INCH, 7*M_INCH
.FormatPageTagEnd
```

The above example shows how to modify the properties of a page.

---

{button ,AL('VENTURA\_FormatCrossReference\_Menu;vent\_format;;;','0,"Defaultoverview",)} **Related Topics**

## **.FormatOverrideCounter (VENTURA)**

**.FormatOverrideCounter .IncrementFromLast=*Boolean*, .StartNumber=*long*, .NumberStyle=*long***

This command overrides the selected counter.

Syntax	Description
<b>.IncrementFromLast</b>	Set to TRUE (-1) to increment from last. Set to FALSE (0) to restart.
<b>.StartNumber</b>	Specifies the starting number to restart. This parameter is only valid when <b>IncrementFromLast</b> is set to FALSE.
<b>.NumberStyle</b>	Specifies the numbering style. The style can be: 0 (1,2,3) 1 (A,B,C) 2 (a,b,c) 3 (I, II, III) 4 (i, ii, iii) 5 (One, Two, Three) 6 (ONE, TWO, THREE) 7 (one, two, three)

### **Example**

```
.FormatOverrideCounter FALSE, 5, 3
```

This example overrides the selected counter using a start number of 5.

---

{button ,AL('VENTURA\_FormatCrossReference\_Menu;vent\_format;;;',0,"Defaultoverview"),} [Related Topics](#)

## .FormatOverrideCounterGet (VENTURA)

**.FormatOverrideCounterGet** .IncrementFromLast=*Boolean* , .StartNumber=*long* , .NumberStyle=*long*

This command returns the settings of the selected counter.

Syntax	Description
.Vertical	Specifies the variable that is assigned the orientation of the specified guideline: TRUE (-1) for vertical guidelines; FALSE (0) for horizontal guidelines.
.IncrementFromLast	Specifies the variable that is assigned TRUE (-1) if the counter increments from last value; otherwise FALSE (0).
.StartNumber	Specifies the variable that is assigned the counter restart number. This parameter is only valid when <b>IncrementFromLast</b> is set to FALSE.
.NumberStyle	Specifies the variable that is assigned the numbering style. The style can be: 0 (1,2,3) 1 (A,B,C) 2 (a,b,c) 3 (I, II, III) 4 (i, ii, iii) 5 (One, Two, Three) 6 (ONE, TWO, THREE) 7 (one, two, three)

### Note

**I** The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example

```
DIM Increment as Boolean
.FormatOverrideCounterGet Increment, StartNum&, NumStyle&
```

This example returns the settings of the current counter.

---

{button ,AL('VENTURA\_FormatCrossReference\_Menu;vent\_format;;;',0,"Defaultoverview"),} Related Topics


**.FormatMergeTags (VENTURA)**

**.FormatMergeTags** *.FileName=string, .TagType=long, .TagList=string, .Options=long*

This command adds tags from other stylesheets to the current stylesheet.

Syntax	Description
.FileName	Specifies the name and path of the stylesheet to add tags from.
.TagType	Specifies the type of tag: 1 Paragraph tag 2 Frame tag 3 Border tag 4 Character tag
.TagList	Specifies the tags to add from the specified stylesheet. Use an empty string ("") to specify all the tags. To specify individual tags, separate the tag names with semi-colons. For example, "TagA;TagB;TagC". This parameter is case-sensitive.
.Options	Specifies the option to use if a tag name being added already exists in the current stylesheet. 1 Does not add a specified tag to the current stylesheet. 2 Overwrites a the tag in the active style sheet (default if omitted). 3 Automatically renames the tag being added to the current style sheet. For example, Body Text could become Body Text_1.

**Note**

 This command corresponds to the Merge Tags button on the Manage Tag List dialog box in Corel VENTURA. Click Format, Manage Tag List, Merge Tags.

**Example**

`.FormatMergeTags "C:\Ventura5\Ssheets\Logos.sty", 1, "", 3`

The above example merges the paragraph tags from the Corel VENTURA 5 LOGOS stylesheet into the active stylesheet. The added tags are renamed if the same tag names currently exist in the active stylesheet.


## .FormatCreateFrame (VENTURA)

**.FormatCreateFrame** *.Left=long, .Top=long, .Width=long, .Height=long, .FrameType=long, .AnchorName=string, .PlantCaret=Boolean*

This command creates a frame on the active document page. Text and graphics can then be inserted into the frame by using the [.FileImportText](#) [.FileImportPicture](#) commands.

Syntax	Description
<b>.Left</b>	Specifies the distance from the left side of the frame's bounding box to the left side of the <a href="#">base page frame</a> , in tenths of micron. Negative values indicate the left side is to the left of the base page frame.
<b>.Top</b>	Specifies the distance from the top of the frame's bounding box to the top side of the <a href="#">base page frame</a> , in tenths of micron. Negative values indicate the top side is above the top of the base page frame.
<b>.Width</b>	Specifies the frame's bounding box width in tenths of a micron.
<b>.Height</b>	Specifies the frame's bounding box height in tenths of a micron.
<b>.FrameType</b>	Specifies the shape of the frame: 0 Rectangle (default if omitted) 1 Inverted rectangle 2 Rounded rectangle 3 Left trapezoid 4 Right trapezoid 5 L90 6 L270 7 Heart 8 Octagon 9 Tear drop 10 Star 11 Bubble 12 Ellipse 13 Diamond
<b>.AnchorName</b>	Specifies the frame's anchor name for anchoring purposes.
<b>.PlantCaret</b>	Specifies whether to plant the text caret in the newly created frame. Set TRUE to plant. FALSE is the default if omitted. This parameter was added in Corel VENTURA 8.

### Note

 This command corresponds to the frame tools on the toolbox in Corel VENTURA.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9
.FrameFirstLine
.TypeText "Corel SCRIPT"
```

The above example creates a tear drop frame with a bounding box of 2.5 by 4.3 inches on the active page. The bounding box is 2 inches from the left and top of the page. The insertion point is placed at the beginning of the first line of the frame and the text Corel SCRIPT is then inserted.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_FormatFrames;;;",0,"Defaultoverview"),} [Related Topics](#)

## .FormatFrameAnchor (VENTURA)

**.FormatFrameAnchor** .AnchoredFrame=*Boolean*, .Position=*long*, .Alignment=*long*, .BaseLineShift=*long*, .Name=*string*

This command sets the frame anchor attributes for the selected frame or changes the attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.AnchoredFrame	Specifies whether to anchor a frame to a location in the text so that it moves whenever the text does. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If this parameter is set to FALSE, all other parameters are ignored.
.Position	Specifies the frame's position relative to the anchor symbol. 1 In-Line (in line with the text containing the anchor symbol) 2 Above Current Line (directly above the anchor symbol) 3 Below Current Line (directly below the anchor symbol) 4 Column Top (at the top of the text column containing the anchor symbol) 5 Column Bottom (at the bottom of the text column containing the anchor symbol) 6 Outside Column (outside the column of text containing the anchor symbol; the frame may still be inside the page margins) 7 Outside Frame (outside the page margins) 8 Within Paragraph (next to the paragraph containing the anchor symbol) 9 Same Page (at the same relative location on the page so that when text moves to a new page, the frame occupies the same location on the new page as it did on the old one)
.Alignment	Specifies the frame's horizontal placement relative to the anchor symbol. Which options are available depends on the value of the <b>.Position</b> parameter. 1 Near binding 2 Away from binding 3 Left 4 Center 5 Right 6 Above/Below at Frame X-Pos
.BaseLineShift	Specifies the frame's vertical position relative to the baseline of the text containing the anchor symbol, in tenths of a micron. Whether or not you can use this parameter depends on the value of the <b>.Position</b> parameter.
.Name	Specifies the name of the frame anchor.

### Note

**I** This command corresponds to the Advanced section of the General tab on the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Advanced.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 2
.FormatFrameAnchor TRUE, 7, 1, 0.1*M_INCH
.FormatObjectEnd
```

The above example enables a frame anchor which uses an Outside Frame position, aligns itself near the binding and uses a baseline shift of 0.1 inches.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0','Defaultoverview',)} [Related Topics](#)



## .FormatFrameAnchorGet (VENTURA)

**.FormatFrameAnchorGet** .AnchoredFrame=*Boolean*, .Position=*long*, .Alignment=*long*, .BaseLineShift=*long*

This function returns the anchor attributes for either the selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
.AnchoredFrame	Specifies the numeric variable that is assigned a value corresponding to whether an anchor is enabled: TRUE (-1) if enabled; otherwise FALSE (0).
.Position	Specifies the numeric variable that is assigned the frame's position relative to the anchor symbol attribute. <div><div>1</div>In-Line (in line with the text containing the anchor symbol) <div>2</div>Above Current Line (directly above the anchor symbol) <div>3</div>Below Current Line (directly below the anchor symbol) <div>4</div>Column Top (at the top of the text column containing the anchor symbol) <div>5</div>Column Bottom (at the bottom of the text column containing the anchor symbol) <div>6</div>Outside Column (outside the column of text containing the anchor symbol; the frame may still be inside the page margins) <div>7</div>Outside Frame (outside the page margins) <div>8</div>Within Paragraph (next to the paragraph containing the anchor symbol) <div>9</div>Same Page (at the same relative location on the page so that when text moves to a new page, the frame occupies the same location on the new page as it did on the old one)</div>
.Alignment	Specifies the numeric variable that is assigned the frame's horizontal placement position relative to the anchor symbol attribute. <div><div>1</div>Near binding <div>2</div>Away from binding <div>3</div>Left <div>4</div>Center <div>5</div>Right <div>6</div>Above/Below at Frame X-Pos</div>
.BaseLineShift	Specifies the frame's vertical position relative to the baseline of the text containing the anchor symbol, in tenths of a micron. Whether or not you can use this parameter depends on the value of the <b>.Position</b> parameter.
.Name	Specifies the name of the frame anchor.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM AF AS BOOLEAN
.FormatFrameAnchorGet AF, , , BLS&
```

In the above example, the function assigns the selected frame's anchor setting and the baseline shift settings to the variables **AF** and **BLS**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"
.FormatFrameAnchorGet AF, , , BLS&
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;','0','Defaultoverview'),} [Related Topics](#)



## **.FormatFrameColGutterOutline (VENTURA)**

**.FormatFrameColGutterOutline** .PageType=*long*, .Column=**long**, .Width=*long*, .Type=*long*, .EndCaps=*long*, .JoinType=*long*, .Aspect=*long*, .Angle=*long*, .DotDash=*long*, .RightArrow=*long*, .LeftArrow=*long*, .BehindFill=*Boolean*


This command sets the column rules attributes for the selected frame or changes the column rules attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.


To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.


To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.PageType	Specifies the pages to apply the column settings to: 1 Left Pages 2 Right Pages 3 All Pages (this is the default if the .PageType parameter is omitted.)
.Column	Specifies the column to apply settings to. The first column is 0, the second column is 1, and so on.
.Width	Specifies the width of the specified ruling line in tenths of a micron.
.Type	Specifies the style of the specified ruling line: 0 None 1 Solid 2 Dotted or Dashed If this parameter is set to Dotted or Dashed, the <b>.DotDash</b> parameter determines the style.
.EndCaps	Specifies the line cap style: 0 Butt 1 Round 2 Square
.JoinType	Specifies the corner style: 0 Miter 1 Round 2 Bevel
.Aspect	Specifies the width of the nib. A square nib has a value of 100; smaller values make the nib narrower, creating a calligraphy effect. Valid values range from 1 to 100, inclusive.
.Angle	Specifies the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree). This option creates a calligraphic effect.
.DotDash	Specifies the style of the specified ruling line if the <b>.Type</b> parameter is set to Dotted or Dashed. The styles correspond to the Style list box in VENTURA's Outline Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
.RightArrow	Specifies the style of the right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.LeftArrow	Specifies the style of the left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.BehindFill	Specifies whether the specified ruling line is placed behind, or in front of, the object's fill. When placed behind, only half the specified ruling line's thickness is visible. This option is particularly useful for outlined text. Set to TRUE (-1) to enable the Behind Fill option; otherwise, set to FALSE (0), which is the default if omitted.

### **Note**

 This command corresponds to the Columns section of the Advanced Columns tab on the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Columns tab, Advanced. Click the pen icon corresponding to the ruling line to edit.

 Using this command, you can only apply formatting changes to one column at a time.

 Use the **.FormatFrameColumns** command to change the number of columns.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 2
.FormatFrameColGutterOutline 3, 2, 0.05*M_INCH, 2, 0, 0, 90, 25, 7
.FormatObjectEnd
```

The above example sets the third ruling line (on all pages) to 0.05 inches with a dashed line. The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.



---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",,)} [Related Topics](#)

## .FormatFrameColGutterOutlineGet (VENTURA)

**.FormatFrameColGutterOutlineGet .PageType=long, .Column=long, .Width=long, .Type=long, .EndCaps=long, .JoinType=long, .Aspect=long, .Angle=long, .DotDash=long, .RightArrow=long, .LeftArrow=long, .BehindFill=Boolean**

This function returns the column rules attributes for a specified column in either the selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
<b>.PageType</b>	Specifies the page setting for the column specified in the <b>.Column</b> parameter: 1 Left Pages 2 Right Pages 3 All Pages
<b>.Column</b>	Specifies the column to return attributes for. The first column is 0, the second column is 1, and so on.
<b>.Width</b>	Specifies the numeric variable that is assigned the width of the specified ruling line in tenths of a micron.
<b>.Type</b>	Specifies the numeric variable that is assigned the style of the specified ruling line: 0 None 1 Solid 2 Dotted or Dashed If this parameter assigns Dotted or Dashed, the <b>.DotDash</b> parameter assigns the specific style.
<b>.EndCaps</b>	Specifies the numeric variable that is assigned the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the numeric variable that is assigned the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the numeric variable that is assigned the width of the nib.
<b>.Angle</b>	Specifies the numeric variable that is assigned the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree).
<b>.DotDash</b>	Specifies the numeric variable that is assigned the style of the specified ruling line if the <b>.Type</b> parameter is set to Dotted or Dashed. The styles correspond to the Style list box in the specified ruling line Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the numeric variable that is assigned the style of the right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the numeric variable that is assigned the style of the left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies the numeric variable that is assigned a value indicating whether the Behind Fill option is enabled: TRUE (-1) if enabled; otherwise, FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatFrameColGutterOutlineGet , 4, Owidth%, Otype%, , , , Odotdash%
```

The above example assigns the attributes for the selected frame's fifth ruling line to the variables **Owidth**, **Otype**, and **Odotdash**.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"
```

```
.FormatFrameColGutterOutlineGet , 4, Owidth&, Otype&, , , , , Odotdash&  
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;',0,"Defaultoverview",)} Related Topics

## .FormatFrameColumnGetAt (VENTURA)

**.FormatFrameColumnGetAt .PageType=long, .Column=long, .Width=long, .GutterWidth=long, .RuleWidth=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long**

This function returns the column attributes for a specified column in either the selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
<b>.PageType</b>	Specifies the page setting for the column specified in <b>.Column</b> : 1 Left Pages 2 Right Pages 3 All Pages
<b>.Column</b>	Specifies the column to return attributes for. The first column is 0, the second column is 1, and so on.
<b>.Width</b>	Specifies the numeric variable that is assigned the width of the specified column, in tenths of a micron.
<b>.GutterWidth</b>	Specifies the numeric variable that is assigned the gutter width for the specified column, in tenths of a micron. The last column does not have a gutter.
<b>.RuleWidth</b>	Specifies the numeric variable that is assigned the gutter vertical rule width for the specified column, in tenths of a micron.
<b>.ColorModel</b>	Specifies the numeric variable that is assigned a return value corresponding to specified column's color model: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u>
<b>.Color1</b>	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color3</b>	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color4</b>	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatFrameColumnsGet , 3, W&, GW&, RW&
```

The above example assigns the selected frame's width, gutter width, and rule width for the fourth column to variables **W**, **GW**, and **RW**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"  
    .FormatFrameColumnsGet , 3, W&, GW&, RW&  
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",')} [Related Topics](#)

## **.FormatFrameColumns (VENTURA)**

**.FormatFrameColumns** .PageType=*long*, .Columns=*long*, .FlowInColumn=*long*, .ColumnBalance=*long*, .GutterRuleOverprint=*B oolean*

This command sets the frame column attributes for the selected frame or changes the attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.PageType	Specifies the pages to apply the column settings to: 1 All Pages (default if omitted) 2 Left Pages 3 Right Pages
.Columns	Specifies the number of columns in the frame. The maximum number of columns you can have depends on the size of the page or frame.
.FlowInColumn	Specifies whether text flows left to right or right to left in the columns 1 left to right 2 right to left
.ColumnBalance	Specify whether VENTURA distributes text so that columns align as closely as possible at the bottom. Setting to Default sets <b>.ColumnBalance</b> to whatever has been specified for the entire chapter using the <b>.FormatChapter</b> commands. 0 Off 1 On 2 Default
.GutterRuleOverprint	Specifies whether to have inter-column rules print over the background color. Set to TRUE (-1) to enable this option; otherwise, FALSE (0) which is the default if omitted. This option is relevant only when printing color separations and should be enabled for most work.

### **Note**

**I** This command corresponds to the Advanced section of the Columns tab of the Format dialog box in Corel VENTURA. Click Format, Frame, Columns tab, Advanced.

**I** To apply formatting to a specified column, use the **.FormatFrameColumnSet** command.

### **Example**

```
.FormatObjectBegin 2  
.FormatFrameColumns 3, 3, 1, 2  
.FormatObjectEnd
```

The above example applies the column settings to all pages. Three columns are set, text flows from left to right, and the chapter default setting determines the column balance.

---

{button ,AL('VENTURA\_FormatFrames;;;','0','Defaultoverview'),} [Related Topics](#)




## **.FormatFrameColumnSet (VENTURA)**

**.FormatFrameColumnSet** .PageType=*long*, .Column=*long*, .Width=*long*, .GutterWidth=*long*, .RuleWidth=*long*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*


This command sets the column attributes for the selected frame or changes the column attributes of a frame tag. The settings specified in this command override the defaults that VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.


To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.PageType	Specifies the pages to apply the column settings to: 1 Left Pages 2 Right Pages 3 All Pages (this is the default if the <b>.PageType</b> parameter is omitted.)
.Column	Specifies the column to apply settings to. The first column is 1, the second column is 2, and so on.
.Width	Specifies the width of the specified column, in tenths of a micron.
.GutterWidth	Specifies the gutter width for the specified column, in tenths of a micron. The last column does not have a gutter.
.RuleWidth	Specifies the gutter vertical rule width for the specified column, in tenths of a micron. Setting it to 0 indicates not to use a rule.
.ColorModel	Specifies the color model to use for the rule, if available: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u>  If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### **Note**

 This command corresponds to the Columns section of the Advanced Columns tab on the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Columns tab, Advanced.

 Using this command, you can only apply formatting changes to one column at a time.

 Use the **.FormatFrameColumns** command to change the number of columns.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 2
.FormatFrameColumnSet 3, 2, 2.5*M_INCH, 0.5*M_INCH, 0.1*M_INCH, 5, 200, 10, 10
.FormatObjectEnd
```

The above example sets the second column to 2.5 inches, the gutter width to 0.5 inches, the vertical rule width to 0.1 inches and the vertical rule color to red.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",')} **Related Topics**



# .FormatFrameColumnsGet (VENTURA)

**.FormatFrameColumnsGet** *.PageType=long*, *.Columns=long*, *.FlowInColumn=long*, *.ColumnBalance=long*, *.GutterRuleOverprint=Boolean*

This function returns the column attributes in either the selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
<b>.PageType</b>	Specifies the page setting to return column attributes for: 1 All Pages 2 Left Pages 3 Right Pages
<b>.Columns</b>	Specifies the numeric variable that is assigned the number of columns in the frame.
<b>.FlowInColumn</b>	Specifies the numeric variable that is assigned a value indicating whether text flows left to right, or right to left in the columns 1 left to right 2 right to left
<b>.ColumnBalance</b>	Specifies the numeric variable that is assigned a value indicating whether VENTURA distributes text so that columns align as closely as possible at the bottom. 0 Off 1 On 2 Default (specified for the entire chapter using the <b>.FormatChapter</b> commands)
<b>.GutterRuleOverprint</b>	Specifies the numeric variable that is assigned a value indicating if inter-column rules print over the background color: TRUE (-1) if this option is enable; FALSE (0) otherwise.

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

## Example

```
.FormatFrameColumnsGet , NumCols&, , ColBal&
```

The above example assigns the selected frame's number of columns and column balance settings to the variables **NumCols** and **ColBal**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"  
    .FormatFrameColumnsGet , NumCols&, , ColBal&  
.FormatFrameTagEnd
```

## .FormatFrameGeneral (VENTURA)

**.FormatFrameGeneral** .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long*, .FrameRotation=*long*, .FitTextVertical=*Boolean*, .LockFrame=*Boolean*, .CaptionType=*long*, .Name=*string*, .FrameFlow=*long*, .ContentRotation=*long*


This command sets frame dimensions, position, angle, and text flow attributes for the selected frame or changes the attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Left	Specifies the distance from the left side of the frame's bounding box to the left side of the <u>base page frame</u> , in tenths of a micron.
.Top	Specifies the distance from the top of the frame's bounding box to the top of the <u>base page frame</u> , in tenths of a micron.
.Width	Specifies the width of the frame's bounding box, in tenths of a micron.
.Height	Specifies the height of the frame's bounding box, in tenths of a micron.
.FrameRotation	Specifies an angle to rotate the frame and its contents in tenths of a degree (1 degree = 10 tenths of degree). A negative value rotates counterclockwise. To rotate only the content of the frame, use the <b>.ContentRotation</b> settings.
.FitTextVertical	Specifies whether to stretch the frame vertically until all the text in it shows or the bottom of the page is reached. Set TRUE (-1) to enable this option; otherwise, FALSE (0) which is the default if omitted.
.LockFrame	Specifies whether to lock the frame so that it can't be resized or moved. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.CaptionType	Specifies a position for the frame's caption: 0 None (to remove captions) 1 Above 2 Below 4 Left 8 Right  Caption position specifications can be combined by adding their corresponding values. For example, to specify Above & Below, use 3 (1 + 2).
.Name	Specifies a name for the frame so that it can be referred to it using VENTURA's cross-referencing feature.
.FrameFlow	Specifies the frame flow: 1 Above/Below 2 Flow around frame 3 Text flows left 4 Left, then right 5 Text through frame 6 Text flows right
.ContentRotation	Specifies an angle to rotate the frame contents in tenths of a degree (1 degree = 10 tenths of degree). A negative value rotates counterclockwise. To rotate the frame, use the <b>.FrameRotation</b> settings.

### Note

 This command corresponds to the Advanced section of the General tab on the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, General tab, Advanced.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 2
.FormatFrameColumnSet 0.5*M_INCH, 0.5*M_INCH, 4*M_INCH, 6*M_INCH, 450, , -1, , "MyFrame"
.FormatObjectEnd
```

The above example sets a frame to 4 inches wide by 6 inches high, 0.5 inches away from the left and top sides of the base page frame. The frame and the contents are rotated 45 degrees, the position and size are locked and the frame is named MyFrame.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",,)} **Related Topics**

## .FormatFrameGeneralGet (VENTURA)

**.FormatFrameGeneralGet** .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long*, .FrameRotation=*long*, .FitTextVertical=*Boolean*, .LockFrame=*Boolean*, .CaptionType=*long*, .Name=*string*, .FrameFlow=*long*, .ContentRotation=*long*

This function returns the general dimension, position, angle, and text flow attributes in either a selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
.Left	Specifies the numeric variable that is assigned the distance from the left side of the frame's bounding box to the left side of the <u>base page frame</u> , in tenths of a micron.
.Top	Specifies the numeric variable that is assigned the distance from the top of the frame's bounding box to the top of the <u>base page frame</u> , in tenths of a micron.
.Width	Specifies the numeric variable that is assigned the frame's bounding box width in tenths of a micron.
.Height	Specifies the numeric variable that is assigned the frame's bounding box height in tenths of a micron.
.FrameRotation	Specifies the numeric variable that is assigned the frame and contents rotation in tenths of a degree (1 degree = 10 tenths of degree). A negative value rotates it counterclockwise.
.FitTextVertical	Specifies the numeric variable that is assigned a value indicating whether the option that stretches the frame vertically until all the text in it shows is enabled: TRUE (-1) if enabled; otherwise, FALSE (0).
.LockFrame	Specifies the numeric variable that is assigned the lock frame attribute: TRUE (-1) if this option is enabled; otherwise, FALSE (0).
.CaptionType	Specifies the numeric variable that is assigned the frame's caption position: 0     None (to remove captions) 1     Above 2     Below 4     Left 8     Right  Caption position specifications can be combined by adding their corresponding values. For example, to specify Above & Below, use 3 (1 + 2).
.Name	Specifies the string variable that is assigned the frame's cross-referencing name.
.FrameFlow	Specifies the numeric variable that is assigned the frame's flow. 1     Above/Below 2     Flow around frame 3     Text flows left 4     Left, then right 5     Text through frame 6     Text flows right
.ContentRotation	Specifies the numeric variable that is assigned the frame contents rotation angle in tenths of a degree (1 degree = 10 tenths of degree). A negative value rotates counterclockwise.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatFrameGeneralGet , , W_frame&, H_frame&, , , , FrameName&
```

The above example assigns the selected frame's width, height, and name to the variables **W\_frame**, **H\_frame**, and **FrameName**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"  
    .FormatFrameGeneralGet , , W_frame&, H_frame&, , , , FrameName&  
.FormatFrameTagEnd
```

---

{button ,AL(^VENTURA\_FormatFrames;;;,0,"Defaultoverview",)} Related Topics

## **.FormatFrameMarginsInside (VENTURA)**

**.FormatFrameMarginsInside** *.PageType=long, .Left=long, .Right=long, .Top=long, .Bottom=long, .FlowInFrame=long*


This command sets the frame inside margins attributes for the selected frame or changes the attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
<b>.PageType</b>	Specifies the pages where the inside margins apply to: 1 Left Pages 2 Right Pages 3 All Pages
<b>.Left</b>	Specifies the distance between the left side of the frame and its contents, in tenths of a micron.
<b>.Right</b>	Specifies the distance between the right side of the frame and its contents, in tenths of a micron.
<b>.Top</b>	Specifies the distance between the top of the frame and its contents, in tenths of a micron.
<b>.Bottom</b>	Specifies the distance between the bottom of the frame and its contents, in tenths of a micron.
<b>.FlowInFrame</b>	Specifies where the text starts flowing in a frame which has been rotated or reshaped: 1 Highest 2 Highest left 3 Highest right

### **Note**

 This command corresponds to the Inside Margins section of the Advanced Margins tab of the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Margins tab, Advanced.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FrameFirst 'selects the first frame on the page
.FormatObjectBegin 2
.FormatFrameMarginsInside 2, 0.2*M_INCH, 0.2*M_INCH, 0.25*M_INCH, 0.4*M_INCH
.FormatObjectEnd
```

The above example sets the frame margin of the first frame on the right pages to 0.2, 0.2, 0.25, 0.4 inches for the respective margin parameters.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",)} Related Topics



## **.FormatFrameMarginsInsideGet (VENTURA)**

**.FormatFrameMarginsInsideGet .PageType=long, .Left=long, .Right=long, .Top=long, .Bottom=long, .FlowInFrame=long**

This function returns the inside margin attributes for either a selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
<b>.PageType</b>	Specifies the page setting for the frame: 1 Left Pages 2 Right Pages 3 All Pages
<b>.Left</b>	Specifies the numeric variable that is assigned the distance between the left side of the frame and its contents, in tenths of a micron.
<b>.Right</b>	Specifies the numeric variable that is assigned the distance between the right side of the frame and its contents, in tenths of a micron.
<b>.Top</b>	Specifies the numeric variable that is assigned the distance between the top of the frame and its contents, in tenths of a micron.
<b>.Bottom</b>	Specifies the numeric variable that is assigned the distance between the bottom of the frame and its contents, in tenths of a micron.
<b>.FlowInFrame</b>	Specifies the numeric variable that is assigned the a value corresponding to where the text starts flowing in a frame which has been rotated or reshaped: 1 Highest 2 Highest left 3 Highest right

### **Note**



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### **Example**

```
.FormatFrameMarginsInsideGet , Lmargin& , Tmargin&
```

The above example assigns the selected frame's left margin and top margin to the variables **Lmargin** and **Tmargin**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"  
  .FormatFrameMarginsInsideGet , Lmargin& , Tmargin&  
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",')} [Related Topics](#)

## **.FormatFrameMarginsOutside (VENTURA)**

**.FormatFrameMarginsOutside** .Left=*long*, .Right=*long*, .Top=*long*, .Bottom=*long*


This command sets the frame outside margins attributes for the selected frame or changes the attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Left	Specifies the distance between the left side of the frame and the surrounding text, in tenths of a micron.
.Right	Specifies the distance between the right side of the frame and the surrounding text, in tenths of a micron.
.Top	Specifies the distance between the top of the frame and the surrounding text, in tenths of a micron.
.Bottom	Specifies the distance between the bottom of the frame and the surrounding text, in tenths of a micron.

### **Note**

 This command corresponds to the Outside Margins section of the Advanced Margins tab of the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Margins tab, Advanced.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 2
.FormatFrameMarginsOutside 0.3*M_INCH, 0.3*M_INCH, 0.5*M_INCH, 0.5*M_INCH
.FormatObjectEnd
```

The above example sets the outside frame margins to 0.3 and 0.5 inches for the respective margin parameters.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",)} **Related Topics**



# .FormatFrameMarginsOutsideGet (VENTURA)

**.FormatFrameMarginsOutsideGet** .Left=*long*, .Right=*long*, .Top=*long*, .Bottom=*long*

This function returns the outside margin attributes for either a selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
.Left	Specifies the numeric variable that is assigned the distance between the left side of the frame and the surrounding text, in tenths of a micron.
.Right	Specifies the numeric variable that is assigned the distance between the right side of the frame and the surrounding text, in tenths of a micron.
.Top	Specifies the numeric variable that is assigned the distance between the top of the frame and the surrounding text, in tenths of a micron.
.Bottom	Specifies the numeric variable that is assigned the distance between the bottom of the frame and the surrounding text, in tenths of a micron.

### Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

### Example

```
.FormatFrameMarginsInsideGet , Rmargin& , Bmargin&
```

The above example assigns the selected frame's right margin value and bottom margin value to the variables **Rmargin** and **Bmargin**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"  
    .FormatFrameMarginsInsideGet , Rmargin& , Bmargin&  
.FormatFrameTagEnd
```

## .FormatFramePicture (VENTURA)

**.FormatFramePicture** .Width=*long*, .Height=*long*, .HorzShift=*long*, .VertShift=*long*, .FitInFrame=*Boolean*, .MaintainAspect=*Boolean*

This command sets imported picture attributes for the selected frame or changes the attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Width	Specifies the width of the imported picture (not the frame), in tenths of a micron. To use this option, the <b>.FitInFrame</b> parameter must be set to FALSE (0). If you specify a width that exceeds width of the frame, VENTURA will crop the picture.
.Height	Specifies the height of the imported picture (not the frame), in tenths of a micron. To use this option, the <b>.FitInFrame</b> parameter must be set to FALSE (0). If you specify a height that exceeds height of the frame VENTURA will crop the picture.
.HorzShift	Specifies the distance to move the picture horizontally within the frame, in tenths of a micron. A positive value moves it to the left; a negative value moves it to the right.
.VertShift	Specifies the distance you want to move the picture vertically within the frame, in tenths of a micron. A positive value moves it up; a negative value moves it down.
.FitInFrame	Specifies whether to have the size of the picture change as the size of the frame changes. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0).
.MaintainAspect	Specifies whether to prevent the picture from distorting when it is resized. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0).

### Note

**I** This command corresponds to the Picture tab of the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Picture tab.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 2
.FormatFramePicture 0.5*M_INCH, 0.5*M_INCH, 0.1*M_INCH, 0.2*M_INCH
.FormatObjectEnd
```

The above example sets picture size to 0.5 inches in width and height. The picture is shifted 0.1 inches horizontally and 0.2 inches vertically.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",')} [Related Topics](#)

## .FormatFramePictureGet (VENTURA)

**.FormatFramePictureGet** .Width=*long*, .Height=*long*, .HorzShift=*long*, .VertShift=*long*, .FitInFrame=*Boolean*, .MaintainAspect=*Boolean*

This function returns the picture attributes for either a selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the .FormatFrameTagBegin and .FormatFrameTagEnd commands.

Syntax	Description
.Width	Specifies a numeric variable that is assigned the width of the imported picture (not the frame), in tenths of a micron.
.Height	Specifies a numeric variable that is assigned the height of the imported picture (not the frame), in tenths of a micron.
.HorzShift	Specifies a numeric variable that is assigned the distance the imported picture is moved horizontally within the frame, in tenths of a micron. A positive value is a movement to the left; a negative value is a movement to the right.
.VertShift	Specifies a numeric variable that is assigned the distance that the imported picture is moved vertically within the frame, in tenths of a micron. A positive value is a upward movement; a negative value is a downward movement.
.FitInFrame	Specifies a numeric variable that is assigned a value indicating whether the size of the picture changes as the size of the frame changes: TRUE (-1) if enabled; otherwise FALSE (0).
.MaintainAspect	Specifies a numeric variable that is assigned a value indicating whether the picture is prevented from distorting when it is resized: TRUE (-1) if enabled; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM FIF AS BOOLEAN
.FormatFramePictureGet PicWidth&, PicHeight&, , ,FIF
```

The above example assigns the selected frame's width, height, and Fit in Frame attributes to the variables **PicWidth**, **PicHeight**, and **FIF**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"
.FormatFramePictureGet PicWidth&, PicHeight&, , ,FIF
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",)} Related Topics

## **.FormatFrameTagBegin (VENTURA)**

**.FormatFrameTagBegin** .TagName=*string*, .TagsInSelection=*Boolean*

Together, the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands are required to change a frame tag's attributes. The two commands enclose VENTURA commands that can modify a frame tag. The **.FormatFrameTagBegin** statement is placed before the first frame modifying command and the **.FormatFrameTagEnd** command is placed after the last frame modifying command.

Syntax	Description
.TagName	Specifies the frame tag to modify. The tag must already exist in the document. This parameter is required if the <b>TagsInSelection</b> parameter is omitted.  Multiple tags can be specified by separating them with semicolons. For example, "Tag1; Tag2; Tag3".
.TagsInSelection	Specifies whether to modify the tags in the current selection. Set to TRUE (-1) to enable this option; otherwise FALSE (0) which is the default if omitted. This parameter is mutually exclusive with the <b>TagName</b> parameter and takes precedence when set to TRUE. This parameter was introduced in Corel VENTURA 8.

### **Note**

The following commands can be used to modify a frame tag.

	<b><u>FormatFrameAnchor command</u></b>
	<b><u>FormatFrameColGutterOutline command</u></b>
	<b><u>FormatFrameColumns command</u></b>
	<b><u>FormatFrameColumnSet command</u></b>
	<b><u>FormatFrameGeneral command</u></b>
	<b><u>FormatFrameMarginsInside command</u></b>
	<b><u>FormatFrameMarginsOutside command</u></b>
	<b><u>FormatFramePicture command</u></b>
	<b><u>FormatFrameTypography command</u></b>
	<b><u>FormatFrameVerticalRule command</u></b>
	<b><u>FormatFrameVerticalRuleSet command</u></b>
	<b><u>FormatFrameVertRuleOutline command</u></b>

### **Example**

```
.FormatFrameTagBegin "NewTag"  
    .FormatFrameMarginsInside .....  
    .FormatFrameAnchor .....  
    .FormatFrameTypography .....  
.FormatFrameTagEnd
```

---

{button ,AL("VENTURA\_FormatFrames;;;','0,"Defaultoverview"),} Related Topics

## **.FormatFrameTagEnd (VENTURA)**

### **.FormatFrameTagEnd**

Together, the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands are required to change a frame tag's attributes. The two commands enclose VENTURA commands that can modify a frame tag. The **.FormatFrameTagBegin** statement is placed before the first frame modifying command and the **.FormatFrameTagEnd** command is placed after the last frame modifying command.

#### **Note**

The following commands can be used to modify a frame tag.

	<b><u>FormatFrameAnchor command</u></b>
	<b><u>FormatFrameColGutterOutline command</u></b>
	<b><u>FormatFrameColumns command</u></b>
	<b><u>FormatFrameColumnSet command</u></b>
	<b><u>FormatFrameGeneral command</u></b>
	<b><u>FormatFrameMarginsInside command</u></b>
	<b><u>FormatFrameMarginsOutside command</u></b>
	<b><u>FormatFramePicture command</u></b>
	<b><u>FormatFrameTypography command</u></b>
	<b><u>FormatFrameVerticalRule command</u></b>
	<b><u>FormatFrameVerticalRuleSet command</u></b>
	<b><u>FormatFrameVertRuleOutline command</u></b>

#### **Example**

```
.FormatFrameTagBegin "NewTag"  
    .FormatFrameMarginsInside .....  
    .FormatFrameAnchor .....  
    .FormatFrameTypography .....  
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;','0',"Defaultoverview",,)} [Related Topics](#)

## **.FormatFrameTypography (VENTURA)**

**.FormatFrameTypography** *.Widows=long, .Orphans=long, .FirstBaseLine=long, .PairKerning=long, .VJWithinFrame=long, .VJMax=long, .VJAroundFrame=long, .VJMaxAtTop=long, .VJMaxAtBottom=long*

This command sets the frame character-to-character and line-to-line spacing attributes for the selected frame or changes the attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Widows	Specifies the minimum number of lines that can be isolated at the top of the page. Two is a typical number, since it prevents single isolated lines.
.Orphans	Specify the minimum number of lines that can be isolated at the bottom of the page. Two is a typical number, since this will prevent single isolated lines.
.FirstBaseLine	<p>Specifies where the first line of text begins on a page or in a frame:</p> <ul style="list-style-type: none"><li>0 Cap-Height</li><li>1 Inter-Line</li><li>2 Default</li></ul> <p>Set to 0 to have uppercase letters align flush with the top of the column or frame. Set to 1 to move the text down by an amount equal to its inter-line spacing. Set to 2 to use the chapter-wide settings.</p>
.PairKerning	<p>Specifies whether to enable or disable pair kerning, which moves certain pairs of adjacent characters closer together:</p> <ul style="list-style-type: none"><li>0 On</li><li>1 Off</li><li>2 Default</li></ul> <p>Set to 0 to activate kerning in paragraphs that have automatic kerning enabled. Manually kerned text is not affected by this option. Set to 2 to use the chapter-wide settings.</p> <p>Pair kerning is available for fonts which have built-in kerning information. Type 1 fonts have this information; most other fonts (including TrueType fonts) do not. Any font lacking kerning information can still be manually kerned.</p>
.VJWithinFrame	<p>Specifies how VENTURA adds space to make columns reach the bottom of the page or frame:</p> <ul style="list-style-type: none"><li>0 Off</li><li>1 Feathering</li><li>2 Carding</li><li>3 Default</li></ul> <p>Set to 1 to add whatever space is needed. Set to 2 to add space in increments equal to the interline spacing. Set to 3 to use the chapter-wide settings.</p>
.VJMax	Specifies the maximum amount of space allowed between elements (paragraphs and tables) within the frame during vertical justification, as a tenth of a percentage.
.VJAroundFrame	<p>Specifies whether VENTURA is allowed to move the frame as it vertically justifies pages:</p> <ul style="list-style-type: none"><li>0 Off</li><li>1 Moveable</li><li>2 Fixed</li><li>3 Default</li></ul> <p>Moveable is the best option for most types of documents. Set to Fixed when it's important that the frame stay in the same location on a page. Set to Off to stop VENTURA from adding any space above or below the frame. Set to Default to use the chapter-wide settings.</p>
.VJMaxAtTop	Specifies the maximum amount of space that VENTURA is allowed to add above the frame as it vertically justifies pages, in tenths of a micron.
.VJMaxAtBottom	Specifies the maximum amount of space that VENTURA is allowed to add above frames as it vertically justifies pages, in tenths of a micron.

### **Note**

**I** This command corresponds to the Typography tab of the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Typography tab.





For omitted parameter and default settings, the active chapter's settings are used.

#### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 2
.FormatFrameTypography 2, 3, , , , 1, 0.25*M_INCH, 0.25*M_INCH
.FormatObjectEnd
```

In the above example, a frame has widows set to 2 lines and orphans set 3 lines. The frame is moveable during vertical justification and the maximum amount of space that can be moved at the top and bottom is 0.25 inches.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",)} **Related Topics**

## .FormatFrameTypographyGet (VENTURA)

**.FormatFrameTypographyGet** .Widows=*long*, .Orphans=*long*, .FirstBaseLine=*long*, .PairKerning=*long*, .VJWithinFrame=*long*, .VJMax=*long*, .VJAroundFrame=*long*, .VJMaxAtTop=*long*, .VJMaxAtBottom=*long*

This function returns the typography attributes for either a selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
.Widows	Specifies the numeric variable that is assigned the minimum number of lines that can be isolated at the top of the page.
.Orphans	Specifies the numeric variable that is assigned the minimum number of lines that can be isolated at the bottom of the page. T
.FirstBaseLine	Specifies the numeric variable that is assigned where the first line of text begins: 0 Cap-Height 1 Inter-Line 2 Default
.PairKerning	Specifies the numeric variable that is assigned the frame's kerning attribute: 0 On 1 Off 2 Default
.VJWithinFrame	Specifies the numeric variable that is assigned a value that indicates how VENTURA adds space to make columns reach the bottom of the page or frame: 0 Off 1 Feathering 2 Carding 3 Default
.VJMax	Specifies the numeric variable that is assigned the maximum amount of space allowed between elements (paragraphs and tables) within the frame during vertical justification, as a tenth of a percentage.
.VJAroundFrame	Specifies the numeric variable that is assigned a value that indicates how VENTURA is allowed to move the frame as it vertically justifies pages: 0 Off 1 Moveable 2 Fixed 3 Default
.VJMaxAtTop	Specifies the numeric variable that is assigned the maximum amount of space that VENTURA is allowed to add above the frame as it vertically justifies pages, in tenths of a micron.
.VJMaxAtBottom	Specifies the numeric variable that is assigned the maximum amount of space that VENTURA is allowed to add above frames as it vertically justifies pages, in tenths of a micron.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatFrameTypographyGet NumWid&, NumOrph&, ,PK&
```

The above example assigns a selected frame's widow, orphan and pair kerning settings to the variables **NumWid**, **NumOrph**, and **PK**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"  
    .FormatFrameTypographyGet NumWid&, NumOrph&, ,PK&  
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;','0',"Defaultoverview"),} [Related Topics](#)

## .FormatFrameVerticalRule (VENTURA)

**.FormatFrameVerticalRule** .PageType=*long*, .NumOfRules=*long*, .Overprint=*Boolean*


This command sets the frame vertical rule attributes for the selected frame or changes the attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.


To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.


To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.


Syntax	Description
.PageType	Specifies the frame's page setting: 1 Left Pages 2 Right Pages 3 All Pages (default if omitted)
.NumOfRules	Specifies the number of vertical rules (lines) to appear in the frame.
.Overprint	Specifies whether to enable overprinting of rules. Set to TRUE (-1) to enable this option; otherwise FALSE (0). If this option is enabled, rules print over the background color. This option is relevant only when printing color separations, and should be enabled for most work.

### Note

 This command corresponds to the Vertical Rules section of the Advanced Margin tab on the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Margins tab, Advanced.

 Vertical rulers do not show in frames containing imported pictures.

 You can only apply one vertical rule at a time with this command.

 Use the **.FormatVerticalRuleSet** command to set an individual vertical rule's attributes.

### Example

```
.FormatObjectBegin 2
.FormatFrameVerticalRule 3, 5, -1
.FormatObjectEnd
```

The above example sets 5 vertical rules in a frame. The Overprint option is enabled. These settings are applied to all pages.

---

{button ,AL('VENTURA\_FormatFrames;;;','0','Defaultoverview'),} [Related Topics](#)

## .FormatFrameVerticalRuleGet (VENTURA)

**.FormatFrameVerticalRuleGet** *.PageType=long, .NumOfRules=long, .Overprint=Boolean*

This function returns vertical rule attributes for a frame.

Syntax	Description
<b>.PageType</b>	Specifies the page setting for the vertical rules: 1 Left Pages 2 Right Pages 3 All Pages
<b>.NumOfRules</b>	Specifies the numeric variable that is assigned the number of vertical rules that appear in the frame.
<b>.Overprint</b>	Specifies the numeric variable that is assigned a value indicating whether the Overprint option is enabled: TRUE (-1) if enabled; FALSE (0) otherwise.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM OP AS BOOLEAN
.FormatFrameVerticalRuleGet PT&, NoR&, OP
```

The above example assigns the selected frame's page setting, number of vertical rules, and Overprint option to the variables **PT**, **NoR**, and **OP**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"
.FormatFrameVerticalRuleGet PT&, NoR&, OP
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.FormatFrameVerticalRuleGetAt (VENTURA)**

**.FormatFrameVerticalRuleGetAt .PageType=long, .RuleNumber=long, .Position=long, .RuleWidth=long, .ColorModel=long, .Color1=long, .Color2=long, .Color3=long, .Color4=long**

This function returns the attributes for a specified vertical rule in either a selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands..

Syntax	Description
<b>.PageType</b>	Specifies the page setting for the vertical rule specified in <b>.RuleNumber</b> : 1 Left Pages 2 Right Pages 3 All Pages
<b>.RuleNumber</b>	Specifies the vertical rule's number that the function returns attributes for. The first vertical rule is 0, the second vertical rule is 1, and so on.
<b>.Enable</b>	Specifies the numeric variable that is assigned a value indicating the specified rule's status: TRUE (-1) if enabled; otherwise FALSE (0).
<b>.Position</b>	Specifies the numeric variable that is assigned the position of the vertical rule (measured from the left side of the frame in tenths of a micron).
<b>.RuleWidth</b>	Specifies the numeric variable that is assigned the vertical rule's width, in tenths of a micron.
<b>.ColorModel</b>	Specifies the numeric variable that is assigned a return value corresponding to specified vertical rule's color model: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u>
<b>.Color1</b>	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
<b>.Color2</b>	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color3</b>	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
<b>.Color4</b>	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

### **Note**



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.



Use the **.FormatFrameVertRuleOutlineGet** function for other vertical rule attributes.

### **Example**

**.FormatFrameColumnsGet , 2, Pos&, RuleWidth&, CM&, c1&, c2&, c3&, c4&**

The above example assigns the selected frame's position and rule width for the third vertical rule to the variables **Pos** and **RuleWidth**, respectively. The color model and color values used for the third vertical rule is assigned to the variables **CM** , **c1**, **c2**, **c3**, and **c4**, respectively.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"  
    .FormatFrameColumnsGet , 2, Pos&, RuleWidth&, CM&, c1&, c2&, c3&, c4&  
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",)} [Related Topics](#)


## **.FormatFrameVerticalRuleSet (VENTURA)**

**.FormatFrameVerticalRuleSet** .PageType=*long*, .RuleNumber=*long*, .Position=*long*, .RuleWidth=*long*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*


This command sets the vertical rule attributes for the selected frame or changes the vertical rule attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.


To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.PageType	Specifies the pages to apply the vertical rule settings to: 1 Left Pages 2 Right Pages 3 All Pages (default if omitted)
.RuleNumber	Specifies the vertical rule to apply settings to. The first vertical rule is 0, the second vertical rule is 1, and so on.
.Position	Specifies the position of the specified vertical rule (measured from the left side of the frame's bounding box in tenths of a micron).
.RuleWidth	Specifies the width for the specified vertical rule, in tenths of a micron.
.ColorModel	Specifies the color model to use for the vertical rule: 1 <u>PANTONE Spot Colors</u> 2 <u>CMYK</u> 3 <u>CMYK255</u> 4 <u>CMY</u> 5 <u>RGB</u> (default if omitted) 6 <u>HSB</u> 7 <u>HLS</u> 8 <u>Black and White</u> 9 <u>Grayscale</u> 11 <u>YIQ</u> 12 <u>L*a*b*</u> 14 <u>PANTONE Hexachrome</u>  If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.

### **Note**

 This command corresponds to the Vertical Rules section of the Advanced Margin tab on the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Margins tab, Advanced.

 Use the **.FormatVerticalRule** command to set the number of vertical rules in a frame.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 2
.FormatFrameVerticalRuleSet 3, 1, 3*M_INCH, 0.5*M_INCH, 5, 200, 10, 10
.FormatObjectEnd
```

The above example sets second vertical rule 3 inches from the left side of the frame. The second vertical rule is set to a width of 0.5 inches and has a red color applied to it.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",,)} **Related Topics**





## **.FormatFrameVertRuleOutline (VENTURA)**

**.FormatFrameVertRuleOutline** .PageType=*long*, .RuleNumber=*long*, .Width=*long*, .Type=*long*, .EndCaps=*long*, .JoinType=*long*, .Aspect=*long*, .Angle=*long*, .DotDash=*long*, .RightArrow=*long*, .LeftArrow=*long*, .BehindFill=*Boolean*


This command sets the vertical ruling line attributes for the selected frame or changes the vertical ruling line attributes of a frame tag. The settings for this command override the defaults Corel VENTURA uses for all frames in the chapter, including the base page. To change the chapter defaults, use the **FormatChapter** commands.

To change a frame tag's attributes, this command must be enclosed with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

To change a selected frame's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.PageType	Specifies the pages to apply the vertical rule settings to: 1 Left Pages 2 Right Pages 3 All Pages (default if omitted)
.RuleNumber	Specifies the vertical rule to apply settings to. The first vertical rule is 0, the second vertical rule is 1, and so on.
.Width	Specifies the numeric variable that is assigned the width of the specified ruling line in tenths of a micron.
.Type	Specifies the numeric variable that is assigned the style of the specified ruling line: 0 None 1 Solid 2 Dotted or Dashed If this parameter assigns Dotted or Dashed, the <b>.DotDash</b> parameter assigns the style.
.EndCaps	Specifies the numeric variable that is assigned the line cap style: 0 Butt 1 Round 2 Square
.JoinType	Specifies the numeric variable that is assigned the corner style: 0 Miter 1 Round 2 Bevel
.Aspect	Specifies the numeric variable that is assigned the width of the nib.
.Angle	Specifies the numeric variable that is assigned the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree).
.DotDash	Specifies the numeric variable that is assigned the style of the specified ruling line if the <b>.Type</b> parameter is set to Dotted or Dashed. The styles correspond to the Style list box in the specified ruling line Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
.RightArrow	Specifies the numeric variable that is assigned the style of the right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.LeftArrow	Specifies the numeric variable that is assigned the style of the left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
.BehindFill	Specifies the numeric variable that is assigned a value indicating whether the behind fill option is enabled: TRUE (-1) if enabled; otherwise, FALSE (0).

### **Note**

 This command corresponds to the Vertical Rules section of the Advanced Margin tab on the Frame Properties dialog box in Corel VENTURA. Click Format, Frame, Margins tab, Advanced.

 Use the **.FormatFrameVerticalRule** command to set the number of vertical rules in a frame.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
```

```
.FormatObjectBegin 2  
.FormatFrameVerticalRuleSet 3, 1, 0.05*M_INCH, 2, 0, 0, 90, 25, 7  
.FormatObjectEnd
```

The above example sets the second vertical ruling line (on all pages) to 0.05 inches with a dashed line. The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.



---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",,)} **Related Topics**

## .FormatFrameVertRuleOutlineGet (VENTURA)

**.FormatFrameVertRuleOutlineGet** *.PageType=long, .RuleNumber=long, .Width=long, .Type=long, .EndCaps=long, .JoinType=long, .Aspect=long, .Angle=long, .DotDash=long, .RightArrow=long, .LeftArrow=long, .BehindFill=Boolean*

This function returns the attributes for a specified vertical ruling line in either a selected frame or for a specified frame tag. To specify the frame tag, enclose the function with the **.FormatFrameTagBegin** and **.FormatFrameTagEnd** commands.

Syntax	Description
<b>.PageType</b>	Specifies the page setting for the vertical rule specified in the <b>.RuleNumber</b> parameter: 1 Left Pages 2 Right Pages 3 All Pages
<b>.RuleNumber</b>	Specifies the vertical rule's number that the function returns attributes for. The first vertical rule is 0, the second vertical rule is 1, and so on.
<b>.Width</b>	Specifies the numeric variable that is assigned the width of the specified ruling line in tenths of a micron.
<b>.Type</b>	Specifies the numeric variable that is assigned the style of the specified ruling line: 0 None 1 Solid 2 Dotted or Dashed If this parameter assigns Dotted or Dashed, the <b>.DotDash</b> parameter assigns the specific style.
<b>.EndCaps</b>	Specifies the numeric variable that is assigned the line cap style: 0 Butt 1 Round 2 Square
<b>.JoinType</b>	Specifies the numeric variable that is assigned the corner style: 0 Miter 1 Round 2 Bevel
<b>.Aspect</b>	Specifies the numeric variable that is assigned the width of the nib.
<b>.Angle</b>	Specifies the numeric variable that is assigned the angle of the nib in tenths of a degree (1 degree equals 10 tenths of a degree).
<b>.DotDash</b>	Specifies the numeric variable that is assigned the style of the specified ruling line if the <b>.Type</b> parameter is set to Dotted or Dashed. The styles correspond to the Style list box in the Specified ruling line Pen dialog box. The first style is identified as 0, the second style is identified as 1, and so on.
<b>.RightArrow</b>	Specifies the numeric variable that is assigned the style of the right arrow. Right-arrow types are listed in the right arrow list box of the Outline Pen dialog box. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.LeftArrow</b>	Specifies the numeric variable that is assigned the style of the left arrow. Left-arrow types are listed in the left arrow list box of the Outline Pen Roll-Up. The types are numbered and identified according to their position in the list (from left to right)  the first listed type is identified as 0, the second listed type is identified as 1, and so on.
<b>.BehindFill</b>	Specifies the numeric variable that is assigned a value indicating whether the behind fill option is enabled: TRUE (-1) if enabled; otherwise, FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



Use the **.FormatFrameVerticalRuleGetAt** function for other vertical rule attributes.



This command cannot be recorded.

### Example

```
.FormatFrameColGutterOutlineGet , 4, Owidth&, Otype&, , , , Odotdash&
```

The above example assigns the attributes for the selected frame's fifth ruling line to the variables **Owidth**, **Otype**, and **Odotdash**.

To use this example to return a specified frame tag's attributes (the Sample frame tag), use the following syntax:

```
.FormatFrameTagBegin "Sample"  
    .FormatFrameColGutterOutlineGet , 4, Owidth&, Otype&, , , , , Odotdash&  
.FormatFrameTagEnd
```

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",')} Related Topics

## .FormatGetFrameTag (VENTURA)

ReturnString\$ = .FormatGetFrameTag ( )

This function returns the frame tag name of the selected frame.

Syntax	Description
ReturnString\$	Specifies the string variable that is assigned the name of the selected frame's tag.

### Note



This command cannot be recorded.



You can also use the .FrameTagGetAt command to find the name of a frame tag.

### Example

```
CurrentFrameTag$ = .FormatGetFrameTag ( )
```

In the above example, the frame tag name of the selected frame is assigned to the **CurrentFrameTag** string variable.

---

{button ,AL("VENTURA\_FormatFrames;;;','0,"Defaultoverview"),} Related Topics

## **.FormatSetFrameTag (VENTURA)**

**.FormatSetFrameTag .TagName=string**

This command applies a specified frame tag to the selected frame.

Syntax	Description
<b>.TagName</b>	Specifies the frame tag to apply.

### **Example**

```
.FormatSetFrameTag "Special homemade"
```

The above example applies the Special homemade frame tag to the selected frame.

---

`{button ,AL("VENTURA_FormatFrameVerticalRule_Menu;vent_frame;;;','0,"Defaultoverview"),}` [Related Topics](#)

## .FrameAutoWrap (VENTURA)

### .FrameAutoWrap

This command draws a path around the perimeter of an imported picture allowing the text to flow around the shape of the picture.

#### Note



This command was introduced in Corel VENTURA 8.



This command corresponds to selecting a frame, right-clicking, and clicking Picture, Autowrap.

#### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatCreateFrame 3*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.FileImportPicture "f:\pictures\kevin.cpt", 0, FALSE
.FrameAutoWrap
```

The above example imports a picture file named KEVIN.CPT (Corel PHOTO-PAINT format) into the active chapter. The frame is then wrapped around the imported picture.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",)} Related Topics

## .FrameContinueTextFlow (VENTURA)

**.FrameContinueTextFlow** *.Left=long, .Top=long, .Width=long, .Height=long, .FrameIndex=long, .Before=Boolean, .TargetPageNumber=long*

This command flows the text of the currently selected frame into a newly created frame or an existing frame.

Syntax	Description
<b>.Left</b>	Specifies the left coordinate of the new frame in tenths of a micron. If flowing text into an existing frame, set this parameter to -1.
<b>.Top</b>	Specifies the top coordinate of the new frame in tenths of a micron. If flowing text into an existing frame, set this parameter to -1.
<b>.Width</b>	Specifies the width of the new frame in tenths of a micron. If flowing text into an existing frame, set this parameter to -1.
<b>.Height</b>	Specifies the height of the new frame in tenths of a micron. If flowing text into an existing frame, set this parameter to -1.
<b>.FrameIndex</b>	Specifies the index number of an existing frame to flow text into. For more information about frame index numbers, see the <a href="#">CurrentFrameIndex</a> function.
<b>.Before</b>	Set to TRUE (-1) to continue the flow before the current frame. Set to FALSE (0) to continue the flow after the current frame. The default is FALSE if omitted.
<b>.TargetPageNumber</b>	Specifies the page number (relative to the chapter) to create the new frame. You can also use the following values: 0      current page (default if omitted) -1     next page -2     previous page

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.ViewGoToPage 1
.FrameFirst FALSE, 1
.FrameContinueTextFlow 3*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH, , 0, -1
```

This example flows the text of the currently selected frame into a frame on the first page.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",)} [Related Topics](#)



## **.FrameMakeSpread (VENTURA)**

### **.FrameMakeSpread**

This command makes the selected frame capable of spreading between two pages. It acts as a toggle by turning spreading ON/OFF.

#### **Note**



You must be in page tag view and have a frame selected to use this command.



This command requires the selected frame to be on the Left page.

#### **Example**

```
.ViewMasterPage  
.FrameFirst FALSE, 1  
.FrameMakeSpread
```

This example selects the first frame on the page tag, and makes it capable of spreading between two pages. It acts as a toggle by turning spreading ON/OFF.

---

{button ,AL("VENTURA\_FormatFrames;;;','0,"Defaultoverview"),} [Related Topics](#)

## **.FrameRemoveFromFrame (VENTURA)**

**.FrameRemoveFromFrame** .TextFile=*Boolean*, .Picture=*Boolean*, .OLEObject=*Boolean*

This command removes the components from the currently selected frame(s).

<b>Syntax</b>	<b>Description</b>
.TextFile	Set to TRUE (-1) to remove text file(s). Set to FALSE (0) to keep them which is the default if omitted.
.Picture	Set to TRUE (-1) to remove picture file(s). Set to FALSE (0) to keep them which is the default if omitted.
.OLEObject	Set to TRUE (-1) to remove OLE object(s). Set to FALSE (0) to keep them which is the default if omitted.

### **Example**

`.FrameRemoveFromFrame FALSE, TRUE, FALSE`

This example removes the picture file from the selected frame.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",,)} [Related Topics](#)

## **.FrameShowOnFacingPage (VENTURA)**

### **.FrameShowOnFacingPage**

This command shows the current frame on the facing page.

#### **Note**



You must be in page tag view and have a frame selected to use this command.

#### **Example**

```
.ViewMasterPage  
.FrameFirst FALSE, 1  
.FrameShowOnFacingPage
```

This example shows the selected frame on the facing page.

---

`{button ,AL('VENTURA_FormatFrames;;;','0,"Defaultoverview",)} Related Topics`

## **.FrameSizeToObject (VENTURA)**

### **.FrameSizeToObject**

This command resizes the frame to fit the OLE object or picture contained in the frame.

#### **Example**

`.FrameSizeToObject`

This example resizes the frame to fit the OLE object or picture contained in the frame.

---

`{button ,AL('VENTURA_FormatFrames;;;',0,"Defaultoverview"),}` [Related Topics](#)

## .FrameTagAddNew (VENTURA)

**.FrameTagAddNew** **.TagName**=*string*, **.CopyFrom**=*string* **.CopyFromFrame**=*Boolean*

This command adds a new frame tag to the active style sheet. The new tag is based on an existing frame tag.

Syntax	Description
<b>.TagName</b>	Specifies the name of the new frame tag. The tag name must not already exist in the style sheet.
<b>.CopyFrom</b>	Specifies the frame tag that the new frame tag is based on.
<b>.CopyFromFrame</b>	Specifies whether to create the tag based on the currently selected frame. Set to TRUE (-1) to enable; otherwise set to FALSE (0). The default is FALSE.

### Note



This command corresponds to the Tag Window. Click Tools, Tag Window.

### Example

```
.FrameTagAddNew "Thickest", "Thicker", FALSE
```

The above example adds a frame tag named Thickest to the active style sheet. The new tag is based on the Thicker tag.

---

{button ,AL("VENTURA\_FormatFrames;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.FrameTagCopy (VENTURA)**

**.FrameTagCopy .TagName=string**

This command copies a frame tag from the active style sheet to the clipboard. This command can be used to copy tags into VENTURA Libraries.

Syntax	Description
<b>.TagName</b>	Specifies the name of the frame tag to copy.

### **Note**



This command corresponds to the Tag Window. Click Tools, Tag Window.

### **Example**

```
.FrameTagCopy "TradeMark"  
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste
```

The above example copies the frame tag TradeMark, creates a VENTURA library named MYLIB.VLB, and pastes the TradeMark tag into the new library.

---

{button ,AL('VENTURA\_FormatFrames;;;','0',"Defaultoverview",)} [Related Topics](#)


## .FrameTagCount (VENTURA)

**ReturnValue& = .FrameTagCount ( )**

This function returns the number of frame tags in the active style sheet.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the number of frame tags in the active style sheet.

### Note

 This command cannot be recorded.

### Example

```
Fcounts& = .FrameTagCount ( )
```


The number of frame tags is assigned the **Fcounts** variable.


---

{button ,AL("VENTURA\_FormatFrames;;;','0,"Defaultoverview",,)} [Related Topics](#)

## .FrameTagDelete (VENTURA)



**.FrameTagDelete** **.TagName=string**, **.ReformatTagName=string**

This command deletes a frame tag from the active style sheet. Any item in the document  and in other documents that are linked to the same style sheet through the Library

 formatted with the deleted tag will be reformatted using another tag of your choice.

Syntax	Description
<b>.TagName</b>	Specifies the name of the frame tag to delete.
<b>.ReformatTagName</b>	Specifies the frame tag to apply to frames tagged with <b>.TagName</b> .

### Note

-  This command corresponds to the Tag Window. Click Tools, Tag Window.
-  You cannot delete the Default frame tag.

### Example

```
.FrameTagDelete "Cats" , "Dogs"
```

The above example deletes the frame tag Cats and applies the Dogs tag to frames using the Cats tag.

---

{button ,AL('VENTURA\_FormatFrames;;;',0,"Defaultoverview"),} [Related Topics](#)



## .FrameTagGetAt (VENTURA)

**ReturnString\$ = .FrameTagGetAt .TagIndex=*long***

This function returns the frame tag name associated with a frame tag index number in the active style sheet.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of a frame tag.
<b>.TagIndex</b>	Specifies the index number of a frame tag. Index numbers are not associated with tag order in the Tag Window. Instead, index numbers are associated by the order in which the frame tags are created. The first tag created is 1, the second tag created is 2, and so on. The last created tag is equal to the return value from the <a href="#">.FrameTagCount</a> function. If a tag is deleted, the tag index is recompiled.

### Note



This command cannot be recorded.



You can also use the [.FormatGetFrameTag](#) command to find the name of a frame tag.

### Example

```
LastTag& = .FrameTagCount ( ) 'number of tags  
LastTagName$ = .FrameTagGetAt (LastTag&)
```

In the above example, the first statement counts the number of frame tags. The second statement is assigned the name of last created frame tag still available in the active style sheet.

---

{button ,AL('VENTURA\_FormatFrames;;;','0,"Defaultoverview",,)} [Related Topics](#)

## **.FrameTagRename (VENTURA)**

**.FrameTagRename** *.OldTagName=string, .NewTagName=string*

This command renames a specified frame tag. Renaming tags may affect the formatting of other documents that are linked to the same style sheet through the Library.

Syntax	Description
<b>.OldTagName</b>	Specifies the frame tag to be renamed.
<b>.NewTagName</b>	Specifies the frame new tag name.

### **Note**



This command corresponds to the Tag Window. Click Tools, Tag Window.



You cannot delete the Default frame tag.

### **Example**

`.FrameTagRename "Name 1", "Name 2"`

The above example renames the frame tag Name 1 to Name 2.

---

`{button ,AL('VENTURA_FormatFrames;;;','0',"Defaultoverview"),}` [Related Topics](#)

## **.InsertFrameAnchor (VENTURA)**

**.InsertFrameAnchor** .Position=*long*. PointX=*long* .PointY=*long*

This command anchors frames to text at the current caret position or given page coordinates. You choose the frame to anchor by selecting it, and then recording the selection with the [.InsertFrameAnchorBegin](#) command.

Syntax	Description
.Position	Specifies where you want the free frame positioned relative to the anchor in the text: <div><div>1</div>In-line</div> <div><div>2</div>Above current line</div> <div><div>3</div>Below current line</div> <div><div>4</div>Column top</div> <div><div>5</div>Column bottom</div> <div><div>6</div>Outside column</div> <div><div>7</div>Outside frame</div> <div><div>8</div>Within paragraph</div> <div><div>9</div>Same page</div>
.PointX	Specifies the horizontal text location in tenths of a micron to anchor relative to the page. When ignored, the text is anchored at the current caret position.
.PointY	Specifies the vertical text location to anchor relative to the page. When ignored, the text is anchored at the current caret position.

### **Note**

**I** This command is only valid after a [.InsertFrameAnchorBegin](#) command.

### **Example**

```
.FileNew 'new file created
.PageFirstLine 'set insertion point to text mode
.TypeText "Some sample text to demonstrate frame anchors." 'inserts text
.FormatCreateFrame FROMINCHES(2.0), FROMINCHES(2.0), FROMINCHES(1.0), FROMINCHES(1.0) 'creates
frame
.FrameFirst FALSE, 1, TRUE 'selects the created frame
.InsertFrameAnchorBegin 'records the selected frame as frame to anchor
.PageFirstLine 'sends insertion point to beginning of first line
.TextWordRight 6, FALSE 'moves insertion point before "anchors"
.InsertFrameAnchor 1 'frame is anchored as in-line
```

The above example demonstrates frame anchoring. See the script comments (after the apostrophes) for more information.

---

{button ,AL('VENTURA\_FormatFrames;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.InsertFrameAnchorBegin (VENTURA)**

### **.InsertFrameAnchorBegin**

This command records the currently selected frame as the frame to be anchored. Use the [.InsertFrameAnchor](#) command to anchor a frame.

#### **Example**

```
.FileNew 'new file created
.PageFirstLine 'set insertion point to text mode
.TypeText "Some sample text to demonstrate frame anchors." 'inserts text
.FormatCreateFrame FROMINCHES(2.0), FROMINCHES(2.0), FROMINCHES(1.0), FROMINCHES(1.0) 'creates
frame
.FrameFirst FALSE, 2, TRUE 'selects the created frame
.InsertFrameAnchorBegin 'records the selected frame as frame to anchor
.PageFirstLine 'sends insertion point to beginning of first line
.TextWordRight 6, FALSE 'moves insertion point before "anchors"
.InsertFrameAnchor 1 'frame is anchored as in-line
```

The above example demonstrates frame anchoring. See the script comments (after the apostrophes) for more information.

---

{button ,AL('VENTURA\_FormatFrames;;;',0,"Defaultoverview",)} [Related Topics](#)

## .CurrentFrameIndex (VENTURA)

**ReturnValue& = .CurrentFrameIndex ( )**

This function returns the index number of the currently selected frame on a page. Index numbers correspond to the order in which the frames are created. The first frame created is 1, the second frame created is 2, and so on. The last created frame is equal to the return value from the **FrameCount** function (with its parameter set to 0). If a frame is deleted, the index numbers are recompiled.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the current frame's index number. Special selected frames are assigned the following frame index numbers: -1     base frame -2     header frame -3     footer frame -4     footnote frame

### Note

 This function was introduced in Corel VENTURA 8.

### Example

```
FC& = .CurrentFrameIndex ( )
```

In the above example, the **FC** variable is assigned the current frame's index number.

---

{button ,AL('VENTURA\_FormatCreateFrame\_Menu;vent\_framenav;;;','0,"Defaultoverview",')} [Related Topics](#)


# .FrameCount (VENTURA)

**ReturnValue**& = **.FrameCount** **.IncludeSpecialFrames**=*Boolean*

This function returns the number of frames on the current page.

Syntax	Description
<b>ReturnValue</b> &	Specifies the numeric variable that is assigned the number of frames.
<b>.IncludeSpecialFrames</b>	Specifies which frames to count. Set to TRUE (-1) to count all the frames on the page. Set to FALSE (0) to exclude the base frame, header frame, footer frame, and footnote frame from the count. If omitted, this parameter is set to FALSE (0). This parameter was introduced in Corel VENTURA 8.

## Note



This function cannot be recorded.

## Example

```
FC& = .FrameCount ( )
```

In the above example, the **FC** variable is assigned the number of frames on the current page (excluding the base frame, header frame, footer frame, and footnote frame).

```
FC_all& = .FrameCount (-1)
```

In the above example, the **FC\_all** variable is assigned the number of frames on the current page.

---

{button ,AL('VENTURA\_FormatCreateFrame\_Menu;vent\_framenav;;;','0,"Defaultoverview",)} Related Topics

## .FrameFirst (VENTURA)

**.FrameFirst** .BaseFrame=*Boolean*, .FrameIndex=*long*, .IgnoreGroup=*Boolean*, .Extend=*Boolean*

This command selects the first frame, the base page frame, or a specified frame on the current page.

Syntax	Description
.BaseFrame	Specifies whether to select the base frame. Set to TRUE (-1) to select the base frame; otherwise, set to FALSE (0). If omitted, set to FALSE.
.FrameIndex	<p>Specifies the index number of a frame. Index numbers correspond to the order in which the frames are created. The first frame created is 1, the second frame created is 2, and so on. The last created frame is equal to the return value from the <a href="#">.FrameCount</a> function (with its parameter set to 0). If a frame is deleted, the index numbers are recompiled.</p> <p>If this parameter is omitted and <b>.BaseFrame</b> is set to FALSE, the first frame created on a page is selected. This parameter is ignored if <b>.BaseFrame</b> is set to TRUE.</p> <p>In Corel VENTURA 8 you can use this parameter to select a specific frame on the current page:</p> <ul style="list-style-type: none"><li>-1 base page frame</li><li>-2 header frame</li><li>-3 footer frame</li><li>-4 footnote frame</li></ul>
.IgnoreGroup	If the frame being selected is part of a group of frames, specifies whether to select the individual frame or the frame group. Set to TRUE (-1) to select the individual frame; otherwise, set to FALSE (0) to select the entire group which is the default if omitted.
.Extend	If frames are currently selected, specifies whether to extend the selection to the frame being selected or to deselect the currently selected frames. Set to TRUE (-1) to extend the selection. Set to FALSE (0) to deselect which is the default if omitted.

### Note



The **.IgnoreGroup** and **.Extend** parameters cannot both be set TRUE (-1).

### Examples

```
.FrameFirst TRUE
```

The above example selects the base frame on the active page.

```
FC& = .FrameCount ( )  
.FrameFirst FALSE, FC
```

In the above example, the **FC** variable is assigned the number of frames on the current page. The second statement selects the last created frame on the current page.

---

{button ,AL("VENTURA\_FrameFirst\_Menu;vent\_framenav;;;",0,"Defaultoverview"),} [Related Topics](#)

## **.FrameFirstLine (VENTURA)**

### **.FrameFirstLine**

This command sends the insertion point to the beginning of the first line of the active text frame.

#### **Note**



You can send the insertion point to the last line using the [.FrameLastLine](#).

#### **Example**

```
.FormatCreateFrame 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 9  
.FrameFirstLine  
.TypeText "Corel SCRIPT"
```

The above example creates a frame and then send the insertion point to the beginning of the first line of the frame. The text Corel SCRIPT is then inserted.

---

{button ,AL('VENTURA\_FrameFirstLine\_Menu;vent\_framenav;;;','0,"Defaultoverview",')} [Related Topics](#)



## **.FrameFirstPara (VENTURA)**

### **.FrameFirstPara**

This command sends the insertion point to the beginning of the first paragraph of the active text frame.

#### **Note**



You can send the insertion point to the first paragraph using the [.FrameLastPara](#).

#### **Example**

`.FrameFirstPara`

---

`{button ,AL('VENTURA_FrameFirstPara_Menu;vent_framenav;;;','0,"Defaultoverview",)} Related Topics`

## **.FrameFooter (VENTURA)**

### **.FrameFooter**

This command selects the footer frame on the current page.

#### **Note**



You can select the footer frame or the header frame using the [.FrameFootnote](#) and [.FrameHeader](#) commands.

#### **Example**

`.FrameFooter`

---

`{button ,AL('VENTURA_FrameFooter_Menu;vent_framenav;;;','0,"Defaultoverview",,)} Related Topics`

## **.FrameFootnote (VENTURA)**

### **.FrameFootnote**

This command selects the footnote frame on the current page.

#### **Note**



You can select the footer frame or the header frame using the [.FrameFooter](#) and [.FrameHeader](#) commands.

#### **Example**

`.FrameFootnote`

---

`{button ,AL('VENTURA_FrameFootnote_Menu;vent_framenav;;;','0,"Defaultoverview",)} Related Topics`

## **.FrameHeader (VENTURA)**

### **.FrameHeader**

This command selects the header frame on the current page.

#### **Note**



You can select the footer frame or the header frame using the [.FrameFooter](#) and [.FrameFootnote](#) commands.

#### **Example**

`.FrameHeader`

---

`{button ,AL('VENTURA_FrameHeader_Menu;vent_framenav;;;','0,"Defaultoverview",,)} Related Topics`

## **.FrameLastLine (VENTURA)**

### **.FrameLastLine**

This command sends the insertion point to the end of the last line of the active text frame.

#### **Note**



You can send the insertion point to the first line using the [.FrameFirstLine](#).

#### **Example**

`.FrameLastLine`

---

`{button ,AL('VENTURA_FrameLastLine_Menu;vent_framenav;;;;','0,"Defaultoverview",,)} Related Topics`

## **.FrameLastPara (VENTURA)**

### **.FrameLastPara**

This command sends the insertion point to the end of the last paragraph of the active text frame.

#### **Note**



You can send the insertion point to the first paragraph using the [.FrameFirstPara](#).

#### **Example**

`.FrameLastPara`

---

`{button ,AL('VENTURA_FrameLastPara_Menu;vent_framenav;;;','0,"Defaultoverview",,)} Related Topics`





# .FrameNext (VENTURA)

**.FrameNext** .IgnoreGroup=*Boolean*, .Extend=*Boolean*

This command selects the next frame ordered after the active frame on the current page. This command is only valid when a frame is currently selected.

Syntax	Description
.IgnoreGroup	If the next frame is part of a group of frames, specifies whether to select the individual frame or the frame group. Set to TRUE (-1) to select the individual frame; otherwise, set to FALSE (0) to select the entire group which is the default if omitted.
.Extend	If frames are currently selected, specifies whether to include the next frame in the selection or to deselect the currently selected frames. Set to TRUE (-1) to include. Set to FALSE (0) to deselect the currently selected frames. FALSE is the default if this parameter is omitted.

## Note

-  Frames are ordered based on the order they are inserted on a page. The base frame is always considered the first frame on a page. Header, footer, and footnote frames are excluded from the ordering.
-  The **.IgnoreGroup** and **.Extend** parameters cannot both be set TRUE (-1).
-  This command corresponds to pressing TAB in VENTURA when a frame is selected.
-  You can use the **.FramePrev** command to select the previous frame on the current page.

## Example

```
.FrameFirst
.FrameNext , TRUE
```

In the above example, the first line selects the first frame on the current page and the second line selects the second frame on the page, extending the selection.

## **.FrameNextLinked (VENTURA)**

### **.FrameNextLinked**

This command selects the next linked frame (if it exists) after the active frame on the current page. The next linked frame can be on a different page from the active frame.

#### **Note**



You can select the previous linked frame with the **.FramePreviousLinked** command.

#### **Example**

`.FrameNextLinked`

---

```
{button ,AL("VENTURA_FrameNextLinked_Menu;vent_framenav;;;",0,"Defaultoverview"),} Related Topics
```







# .FramePrev (VENTURA)

**.FramePrev** .IgnoreGroup=*Boolean*, .IgnoreGroup=*Boolean*

This command selects the previous frame ordered before the active frame on the current page. This command is only valid when a frame is currently selected.

Syntax	Description
.IgnoreGroup	If the previous frame is part of a group of frames, specifies whether to select the individual frame or the frame group. Set to TRUE (-1) to select the individual frame; otherwise, set to FALSE (0) to select the entire group which is the default if omitted.
.Extend	If frames are currently selected, specifies whether to include the previous frame in the selection or to deselect the currently selected frames. Set to TRUE (-1) to include. Set to FALSE (0) to deselect the currently selected frames. FALSE is the default if this parameter is omitted.

## Note

-  Frames are ordered based on the order they are inserted on a page. The base frame is always considered the first frame on a page. Header, footer, and footnote frames are excluded from the ordering.
-  The **.IgnoreGroup** and **.Extend** parameters cannot both be set TRUE (-1).
-  This command corresponds to pressing SHIFT+TAB in VENTURA when a frame is selected. A frame must be selected
-  You can use the **.FrameNext** command to select the next frame on the current page.

## Example

`.FramePrev`

---

{button ,AL("VENTURA\_FramePrev\_Menu;vent\_framenav;;;','0,"Defaultoverview"),} Related Topics

## **.FramePrevLinked (VENTURA)**

### **.FramePrevLinked**

This command selects the previous linked frame (if it exists) before the active frame on the current page. The previous linked frame can be on a different page from the active frame.

#### **Note**



You can select the next linked frame with the [\*\*.FrameNextLinked\*\*](#) command.

#### **Example**

`.FramePrevLinked`

---

```
{button ,AL("VENTURA_FramePrevLinked_Menu;vent_framenav;;;",0,"Defaultoverview"),} Related Topics
```

## **.FrameSelectAll (VENTURA)**

### **.FrameSelectAll**

This command selects all the frames on the current page.

#### **Example**

```
.FrameSelectAll
```

---

```
{button ,AL('VENTURA_FrameSelectAll_Menu;vent_framenav;;;','0,"Defaultoverview",)} Related Topics
```

## **.ClearTransforms (VENTURA)**

### **.ClearTransforms**

This command removes transformations from graphic objects. Transformations include resizing, skewing, and rotation of objects.

#### **Note**



This command cannot be recorded.

#### **Example**

```
.DrawRectangle 653420, 941107, 1081703, 1219793, 0  
.RotateObject 268  
.EditDuplicate  
.ClearTransforms
```

The above example creates a rectangle and then rotates it. The rotated rectangle is then duplicated. The duplicate, which is selected, has the rotation transformation removed from it.

---

{button ,AL("VENTURA\_DrawCurveBegin\_Menu;vent\_graphic;;;",0,"Defaultoverview",)} [Related Topics](#)




# .DrawArtisticText (VENTURA)

**.DrawArtisticText** *.X=long, .Y=long, .Text=string*

This command draws artistic text on the active page.

Syntax	Description
.X	Specifies the distance from the left side of the artistic text's bounding box to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the left side is to the left of the base page frame.
.Y	Specifies the distance from the top of the artistic text's bounding box to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the top side is above the top of the base page frame.
.Text	Specifies the text used in the artistic text.

**Note**

-  This command cannot be recorded.
-  This command corresponds to the Artistic Text tool on the Graphic Tools flyout of the toolbox. This command cannot be recorded.
-  This command draws artistic text using the current artistic text font settings.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawArtisticText 2*M_INCH, 2*M_INCH, "Corel VENTURA"
```

The above example draws artistic text (Corel VENTURA) that has a bounding box that is 2 inches from the left and top of the base page frame.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

## .DrawCallout (VENTURA)

**.DrawCallout .Left=long, .Top=long, .Width=long, .Height=long**

This command draws a callout on the active page. This command is valid only when a curve has been defined by using **DrawCurveBegin**, followed by set of curve drawing commands. **DrawCurveEnd** command should not be used with **DrawCallout**.

Syntax	Description
<b>.Left</b>	Specifies the distance from the left side of the callout's bounding box to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the left side is to the left of the base page frame.
<b>.Top</b>	Specifies the distance from the top of the callout's bounding box to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the top side is above the top of the base page frame.
<b>.Width</b>	Specifies the width (the distance from the left side of the bounding box to right side) of the callout's bounding box, in tenths of micron.
<b>.Height</b>	Specifies the height (the distance from the top of the bounding box to the bottom) of the callout's bounding box, in tenths of micron.

### Example

```
.DrawEllipse 430391, 1160851, 511322, 430972
.DrawCurveBegin 0, 0, 3
.DrawCurveAddNode 937184, 1423817, 32
.DrawCurveAddNode 934409, 1781743, 33
.DrawCurveAddNode 1131633, 1781743, 33
.DrawCallout 1137983, 1730943, 254000, 254000
.FrameFirstLine
.TypeText "This is a circle"
```

This example draws a graphic object (ellipse) and then draws a callout to the graphic object.


---

`{button ,AL(`VENTURA_DrawArtisticText_Menu;vent_graphic;;;;',0,"Defaultoverview",)}` [Related Topics](#)

## **.DrawCurveAddNode (VENTURA)**

**.DrawCurveAddNode .X=*long*, .Y=*long*, .NodeData=*long***




This command is recorded when the Freehand Tool () is used to draw lines. Do not write this command into scripts since the command is designed for recording purposes only. To draw lines by writing commands in a script, use the [.DrawCurveLineTo](#) command.

This command is always part of contiguous block of other **.DrawCurveAddNode** commands. The first **.DrawCurveAddNode** command in a block is immediately preceded by the [.DrawCurveBegin](#) command. The last command in a block must be followed by the [.DrawCurveEnd](#) command.

Syntax	Description
<b>.X</b>	Specifies the distance from a node in the drawn line to the left side of the <a href="#">base page frame</a> , in tenths of micron. Negative values indicate the node is to the left of the base page frame.
<b>.Y</b>	Specifies the distance from a node in the drawn line to the top side of the <a href="#">base page frame</a> , in tenths of micron. Negative values indicate the node is above the top of the base page frame.
<b>.NodeData</b>	Specifies information which the node uses in determining the path to the next node in the drawn line.

### **Example**

```
.DrawCurveBegin 0, 0, 25
.DrawCurveAddNode 349397, 861041, 32
.DrawCurveAddNode 349397, 1076285, 3
.DrawCurveAddNode 515240, 1146856, 3
.DrawCurveAddNode 716369, 1108042, 34
.DrawCurveAddNode 741069, 1019827, 3
.DrawCurveAddNode 762241, 896327, 3
.DrawCurveAddNode 719898, 811641, 34
.DrawCurveAddNode 666969, 811641, 3
.DrawCurveAddNode 564640, 815170, 3
.DrawCurveAddNode 525826, 850455, 38
.DrawCurveAddNode 405854, 959841, 3
.DrawCurveAddNode 568169, 1213899, 3
.DrawCurveAddNode 476426, 1323285, 38
.DrawCurveAddNode 437611, 1369157, 3
.DrawCurveAddNode 247068, 1393857, 3
.DrawCurveAddNode 211782, 1333871, 38
.DrawCurveAddNode 172968, 1266828, 3
.DrawCurveAddNode 225896, 1097456, 3
.DrawCurveAddNode 328225, 1129213, 38
.DrawCurveAddNode 381154, 1146856, 3
.DrawCurveAddNode 405854, 1168028, 3
.DrawCurveAddNode 434083, 1210371, 38
.DrawCurveAddNode 511712, 1333871, 3
.DrawCurveAddNode 716369, 1284471, 3
.DrawCurveAddNode 832813, 1284471, 34
.DrawCurveEnd
```

The above example is recorded using the Freehand tool. Click  to see what this script draws.





---

{button ,AL('VENTURA\_DrawArtisticText\_Menu;vent\_graphic;;;',0,"Defaultoverview"),} [Related Topics](#)

# .DrawCurveBegin (VENTURA)

**.DrawCurveBegin** .X=*long*, .Y=*long*, .Points=*long*

This command initiates the drawing of line and curve graphic objects. Use the **.DrawCurveEnd** command to terminate a sequence of drawing commands. The following commands can be used to draw line and curve graphic objects:

-  **.DrawCurveCurveTo**
-  **.DrawCurveAddNode**
-  **.DrawCurveLineTo**
-  **.DrawCurveMoveTo**

This command uses its parameters differently depending on the drawing sequence it is initiating.

Syntax	Description
.X	<p>Using with <b>.DrawCurveCurveTo</b> command: Specifies the distance from the curve's horizontal start point to the left side of the <u>base page frame</u>, in tenths of micron. Negative values indicate the point is to the left of the base page frame.</p> <p>Using with <b>.DrawCurveAddNode</b> command: This parameter is ignored.</p> <p>Using with <b>.DrawCurveLineTo/.DrawCurveMoveTo</b> commands: Specifies the distance from the first point in the line to the left side of the <u>base page frame</u>, in tenths of micron. Negative values indicate the point is to the left of the base page frame.</p>
.Y	<p>Using with <b>.DrawCurveCurveTo</b> command: Specifies the distance from the curve's vertical start point to the top side of the <u>base page frame</u>, in tenths of micron. Negative values indicate the point is above the top of the base page frame.</p> <p>Using with <b>.DrawCurveAddNode</b> command: This parameter is ignored.</p> <p>Using with <b>.DrawCurveLineTo/.DrawCurveMoveTo</b> commands: Specifies the distance from the first point in the line to the top side of the <u>base page frame</u>, in tenths of micron. Negative values indicate the point is above the top of the base page frame.</p>
.Points	<p>Using with <b>.DrawCurveCurveTo</b> command: This parameter is not used.</p> <p>Using with <b>.DrawCurveAddNode</b> command: Specifies the number of <b>.DrawCurveAddNode</b> commands that follow.</p> <p>Using with <b>.DrawCurveLineTo/.DrawCurveMoveTo</b> commands: This parameter is not used.</p>

## Example

See the drawing commands noted above for examples.

---

{button ,AL('VENTURA\_DrawCurveBegin\_Menu;vent\_graphic;;;',0,"Defaultoverview",)} Related Topics



## **.DrawCurveClosePath (VENTURA)**

### **.DrawCurveClosePath**

This command converts a single-segment or multi-segment curve drawn with the [.DrawCurveCurveTo](#) command into a closed graphic object.

#### **Note**



This command must follow a [.DrawCurveCurveTo](#) command and must precede the [.DrawCurveEnd](#) command.

#### **Example**

See the [.DrawCurveCurveTo](#) command for an example.

---

{button ,AL('VENTURA\_DrawCurveClosePath\_Menu;vent\_graphic;;;','0,"Defaultoverview",')} [Related Topics](#)

## .DrawCurveCurveTo (VENTURA)

**.DrawCurveCurveTo .X1=long, .Y1=long, .X2=long, .Y2=long, .XEnd=long, .YEnd=long**

This command draws a line graphic as a curve. A curve is drawn by specifying a control point (vertex) and fitting the curve to the specified points. This command can be used to draw a single-segment curve or a multiple-segment curve.

To draw a single-segment curve, the **.DrawCurveCurveTo** command must be immediately preceded by the **.DrawCurveBegin** command and followed by the **.DrawCurveEnd** command. The single-segment curve can be converted into a closed graphic object by preceding the **.DrawCurveEnd** command with the **.DrawCurveClosePath** command.

To draw a multi-segment curve, a **.DrawCurveCurveTo** command is used for each segment, and the commands must be in a contiguous block. The first **.DrawCurveCurveTo** command must be immediately preceded by the **.DrawCurveBegin** command. The last **.DrawCurveCurveTo** command is followed by the **.DrawCurveEnd** command. The multi-segment curve can also be converted into a closed graphic object by preceding the **.DrawCurveEnd** command with the **.DrawCurveClosePath** command.

Syntax	Description
<b>.X1</b>	Specifies the distance from the curve's horizontal start point to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the point is to the left of the base page frame.
<b>.Y1</b>	Specifies the distance from the curve's vertical start point to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the point is above the top of the base page frame.
<b>.X2</b>	Specifies the distance from the curve's horizontal end point to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the point is to the left of the base page frame.
<b>.Y2</b>	Specifies the distance from the curve's vertical end point to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the point is above the top of the base page frame.
<b>.XEnd</b>	Specifies the distance from the curve's horizontal control point to the left side of the <u>base page frame</u> , in tenths of micron. Control points act as vertexes that control the shape of the curve. Negative values indicate the point is to the left of the base page frame.
<b>.YEnd</b>	Specifies the distance from the curve's vertical control point to the top side of the <u>base page frame</u> , in tenths of micron. Control points act as vertexes that control the shape of the curve. Negative values indicate the point is above the top of the base page frame.

### Note

**I** If a **.DrawCurveCurveTo** command is immediately preceded by a **.DrawCurveBegin** command, and the **X1** parameter is not equal to the **.DrawCurveBegin** command's **X** parameter, the **X1** parameter value is overridden by the **X** parameter value. The same applies to the **Y1** and **Y** parameters.

**I** If a **.DrawCurveCurveTo** command is immediately preceded by another **.DrawCurveCurveTo** command (multi-segment curve), and the **X1** parameter is not equal to the preceding command's **X2** parameter, the **X1** parameter value is overridden by the preceding command's **X2** parameter value. The same applies to the **Y1** and **Y2** parameters.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawCurveBegin 1*M_INCH, 3.2*M_INCH
.DrawCurveCurveTo 1*M_INCH, 3.2*M_INCH, 4.3*M_INCH, 2.7*M_INCH, 2*M_INCH, 5*M_INCH
.DrawCurveEnd
```


The above example draws a single-segment curve. Click **I** to see what this example draws. The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawCurveBegin 1*M_INCH, 3.2*M_INCH
.DrawCurveCurveTo 1*M_INCH, 3.2*M_INCH, 4.3*M_INCH, 2.7*M_INCH, 2*M_INCH, 5*M_INCH
.DrawCurveClosePath
.DrawCurveEnd
```


This example is the same as the previous except the **.DrawCurveClosePath** command is used to create a closed graphic object. Click **I** to see what this example draws.

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawCurveBegin 1*M_INCH, 3.2*M_INCH
.DrawCurveCurveTo 1*M_INCH, 3.2*M_INCH, 4.3*M_INCH, 2.7*M_INCH, 2*M_INCH, 5*M_INCH
```

```
.DrawCurveCurveTo 4.3*M_INCH, 2.7*M_INCH, 7*M_INCH, 4*M_INCH, 5*M_INCH, 2*M_INCH
.DrawCurveClosePath
.DrawCurveEnd
```

This example draws a multi-segment curve. Click  to see what this example draws.

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawCurveBegin 1*M_INCH, 3.2*M_INCH
.DrawCurveCurveTo 1*M_INCH, 3.2*M_INCH, 4.3*M_INCH, 2.7*M_INCH, 2*M_INCH, 5*M_INCH
.DrawCurveCurveTo 4.3*M_INCH, 2.7*M_INCH, 7*M_INCH, 4*M_INCH, 5*M_INCH, 2*M_INCH
.DrawCurveClosePath
.DrawCurveEnd
```

This example is the same as the previous except the **.DrawCurveClosePath** command is used to create a closed graphic object. Click  to see what this example draws.

---

{button ,AL("VENTURA\_DrawCurveCurveTo\_Menu;vent\_graphic;;;",0,"Defaultoverview",)} Related Topics

## **.DrawCurveEnd (VENTURA)**

**.DrawCurveEnd** *.ReplaceSelection=Boolean*

This command ends the process drawing the curve and draws it in the document.

Syntax	Description
.ReplaceSelection	Set to TRUE (-1) to replace the current selection with the newly defined curve. Set to FALSE (0) to add a new curve. The default is FALSE.

### **Example**

```
.DrawCurveBegin 123456, 123456
.DrawCurveLineTo 654321, 123456
.DrawCurveLineTo 555555, 111111
.DrawCurveLineTo 444444, 222222
.DrawCurveEnd FALSE
```

This example ends the process of drawing the curve and draws it in the document without replacing the current selection.

---

{button ,AL('VENTURA\_DrawCurveEnd\_Menu;;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.DrawCurveLineTo (VENTURA)**

**.DrawCurveLineTo .X=long, .Y=long**

This command continues the drawing of a line to a specified point.

This command is always part of contiguous block of other **.DrawCurveLineTo** or **.DrawCurveMoveTo** commands. The first **.DrawCurveLineTo** or **.DrawCurveMoveTo** command in a block is immediately preceded by the **.DrawCurveBegin** command which specifies the first point in a line. The last command in a block is followed by the **.DrawCurveEnd** command.

Syntax	Description
<b>.X</b>	Specifies the distance from a point in the line to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the point is to the left of the base page frame.
<b>.Y</b>	Specifies the distance from a point in the line to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the point is above the top of the base page frame.

### **Note**




This command cannot be recorded. However, recording Freehand Tool (




) actions records **.DrawCurveAddNode** commands in a script, which is similar to the **.DrawCurveLineTo** command.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawCurveBegin M_INCH*0.14, M_INCH*0.57
.DrawCurveLineTo M_INCH*0.24, M_INCH*0.65
.DrawCurveLineTo M_INCH*0.29, M_INCH*0.85
.DrawCurveLineTo M_INCH*0.40, M_INCH*1.17
.DrawCurveLineTo M_INCH*0.49, M_INCH*1.18
.DrawCurveLineTo M_INCH*0.55, M_INCH*1.24
.DrawCurveLineTo M_INCH*0.75, M_INCH*2.24
.DrawCurveLineTo M_INCH*0.85, M_INCH*2.43
.DrawCurveLineTo M_INCH*1.00, M_INCH*2.46
.DrawCurveLineTo M_INCH*1.35, M_INCH*2.55
.DrawCurveLineTo M_INCH*1.50, M_INCH*2.67
.DrawCurveLineTo M_INCH*1.32, M_INCH*2.74
.DrawCurveLineTo M_INCH*1.22, M_INCH*2.84
.DrawCurveLineTo M_INCH*0.85, M_INCH*2.99
.DrawCurveLineTo M_INCH*1.01, M_INCH*3.40
.DrawCurveLineTo M_INCH*0.32, M_INCH*3.50
.DrawCurveLineTo M_INCH*0.18, M_INCH*3.61
.DrawCurveLineTo M_INCH*0.08, M_INCH*3.71
.DrawCurveLineTo M_INCH*0.39, M_INCH*3.85
.DrawCurveEnd
```

The above example creates a line. Click  to see what this script example draws. The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

The next example creates the same line, but by using the **.DrawCurveMoveTo** command, breaks are made part of the line. Click  to see what this script example draws.

```
.DrawCurveBegin M_INCH*0.14, M_INCH*0.57
.DrawCurveLineTo M_INCH*0.24, M_INCH*0.65
.DrawCurveLineTo M_INCH*0.29, M_INCH*0.85
.DrawCurveLineTo M_INCH*0.40, M_INCH*1.17
.DrawCurveLineTo M_INCH*0.49, M_INCH*1.18
.DrawCurveLineTo M_INCH*0.55, M_INCH*1.24
.DrawCurveMoveTo M_INCH*0.75, M_INCH*2.24
.DrawCurveLineTo M_INCH*0.85, M_INCH*2.43
.DrawCurveLineTo M_INCH*1.00, M_INCH*2.46
.DrawCurveLineTo M_INCH*1.35, M_INCH*2.55
.DrawCurveLineTo M_INCH*1.50, M_INCH*2.67
.DrawCurveLineTo M_INCH*1.32, M_INCH*2.74
.DrawCurveLineTo M_INCH*1.22, M_INCH*2.84
.DrawCurveMoveTo M_INCH*0.85, M_INCH*2.99
.DrawCurveLineTo M_INCH*1.01, M_INCH*3.40
.DrawCurveLineTo M_INCH*0.32, M_INCH*3.50
.DrawCurveLineTo M_INCH*0.18, M_INCH*3.61
.DrawCurveLineTo M_INCH*0.08, M_INCH*3.71
```

```
.DrawCurveLineTo M_INCH*0.39, M_INCH*3.85  
.DrawCurveEnd
```

---

{button ,AL(^VENTURA\_DrawCurveLineTo\_Menu;vent\_graphic;;;',0,"Defaultoverview"),} Related Topics

## **.DrawCurveMoveTo (VENTURA)**


**.DrawCurveMoveTo .X=*long*, .Y=*long***

This command creates breaks in lines drawn with the **.DrawCurveLineTo** commands.

This command is always part of contiguous block of other **.DrawCurveLineTo** or **.DrawCurveMoveTo** commands. The first **.DrawCurveLineTo** or **.DrawCurveMoveTo** command in a block is immediately preceded by the **.DrawCurveBegin** command which specifies the first point in a line. The last command in a block is followed by the **.DrawCurveEnd** command.

Syntax	Description
<b>.X</b>	Specifies the distance from the point where the drawn line continues to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the point is to the left of the base page frame.
<b>.Y</b>	Specifies the distance from the point where the drawn line continues to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the point is above the top of the base page frame.

### **Note**

 This command cannot be recorded.

### **Example**

See the **.DrawCurveLineTo** command for an example.

---

{button ,AL('VENTURA\_DrawCurveMoveTo\_Menu;vent\_graphic;;;','0',"Defaultoverview"),} Related Topics

## **.DrawEllipse (VENTURA)**

**.DrawEllipse .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long***

This command draws an ellipse graphic on the active page.

Syntax	Description
<b>.Left</b>	Specifies the distance from the left side of the ellipse's bounding box to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the left side is to the left of the base page frame.
<b>.Top</b>	Specifies the distance from the top of the ellipse's bounding box to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the top side is above the top of the base page frame.
<b>.Width</b>	Specifies the width (the distance from the left side of the bounding box to right side) of the ellipse's bounding box, in tenths of micron.
<b>.Height</b>	Specifies the height (the distance from the top of the bounding box to the bottom) of the ellipse's bounding box, in tenths of micron.

### **Note**

 This command corresponds to the Ellipse tool on the Graphic Tools flyout of the toolbox.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_DrawEllipse\_Menu;vent\_graphic;;;','0,"Defaultoverview",)} **Related Topics**



## .DrawPolygon (VENTURA)

**.DrawPolygon .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long*, .Sides=*long***

This command draws a polygon graphic on the active page.

Syntax	Description
<b>.Left</b>	Specifies the distance from the left side of the polygon's bounding box to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the left side is to the left of the base page frame.
<b>.Top</b>	Specifies the distance from the top of the polygon 's bounding box to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the top side is above the top of the base page frame.
<b>.Width</b>	Specifies the width (the distance from the left side of the bounding box to right side) of the polygon 's bounding box, in tenths of micron.
<b>.Height</b>	Specifies the height (the distance from the top of the bounding box to the bottom) of the polygon 's bounding box, in tenths of micron
<b>.Sides</b>	Specifies the number of sides on the polygon.

### Note

 This command corresponds to the Polygon tool on the Graphic Tools flyout of the toolbox.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawPolygon 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH, 6
```

The above example draws a polygon with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame. The polygon has six sides making it a hexagon.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_DrawPolygon\_Menu;vent\_graphic;;;','0,"Defaultoverview",)} Related Topics

## .DrawRectangle (VENTURA)

**.DrawRectangle .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long*, .Roundness=*long***

This command draws a rectangle graphic on the active page.

Syntax	Description
<b>.Left</b>	Specifies the distance from the left side of the rectangle to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the left side is to the left of the base page frame.
<b>.Top</b>	Specifies the distance from the top of the rectangle to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the top side is above the top of the base page frame.
<b>.Width</b>	Specifies the width (the distance from the left side of the rectangle to right side) of the rectangle, in tenths of micron.
<b>.Height</b>	Specifies the height (the distance from the top of the rectangle to the bottom) of the rectangle, in tenths of micron
<b>.Roundness</b>	Specifies the roundness of the rectangle's corners. This parameter accepts values from 0 to 100. The higher the parameter setting, the more round the corners become. If omitted, this parameter is set to 0.

### Note



This command corresponds to the Rectangle tool on the Graphic Tools flyout of the toolbox.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
```

The above example draws a rectangle 3 by 4 inches. The rectangle's edges are 2 inches from the left and top of the base page frame.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_DrawRectangle\_Menu;vent\_graphic;;;','0,"Defaultoverview",')} Related Topics

## .GetGraphicType (VENTURA)

ReturnValue& = .GetGraphicType ( )

This function returns the type for the currently selected graphic.

Syntax	Description
ReturnValue&	Specifies the numeric variable that is assigned a value corresponding to the currently selected graphic. 0      None 1      Unknown/Multiple selection 2      Rectangle 3      Ellipse 4      Polygon 5      Curve/Star 6      Artistic Text 7      Group

### Note



This command cannot be recorded.

### Example

```
.GraphicFirst  
Gtype& = .GetGraphicType ( )
```

In the above example, the first line selects the first graphic on the current page. The selected graphic's type is assigned to the **Gtype** variable.

---

{button ,AL('VENTURA\_GetGraphicType\_Menu;vent\_graphic;;;;',0,"Defaultoverview",,)} [Related Topics](#)

## .GraphicCount (VENTURA)

**ReturnValue& =.GraphicCount ( )**

This function returns the number of graphics on the current page.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the number of graphics on the current page.

### Note



This command cannot be recorded.

### Example

```
NumGraphics& = .GraphicCount ( )
```

In the above example the number of graphics on the current page is assigned to the **NumGraphics** variable.

---

{button ,AL("VENTURA\_GraphicCount\_Menu;vent\_graphic;;;','0,"Defaultoverview"),} [Related Topics](#)



# .GraphicFirst (VENTURA)

**.GraphicFirst** . GraphicIndex=*long*, .IgnoreGroup=*Boolean*, .Extend=*Boolean*

This command selects the first graphic or a specified graphic on the current page.

Syntax	Description
.GraphicIndex	Specifies the index number of a graphic. Index numbers correspond to the order in which the graphics are created. The first graphic created is 1, the second graphic created is 2, and so on. The last created graphic is equal to the return value from the <b>.GraphicCount</b> function. If a graphic is deleted, the index numbers are recompiled.  If this parameter is omitted, the first graphic created on a page is selected.
.IgnoreGroup	If the graphic being selected is part of a group of graphics, specifies whether to select the individual graphic or the graphic group. Set to TRUE (-1) to select the individual graphic; otherwise, set to FALSE (0) to select the entire group which is the default if omitted.
.Extend	If graphics are currently selected, specifies whether to extend the selection to the graphic being selected or to deselect the currently selected graphics. Set to TRUE (-1) to extend the selection. Set to FALSE (0) to deselect which is the default if omitted.

## Note

-  This command cannot be recorded.
-  The **.IgnoreGroup** and **.Extend** parameters cannot both be set TRUE (-1).

## Examples

`.GraphicFirst`

The above example selects first graphic on the active page.

```
GC& = .GraphicCount ( )
.GraphicFirst GC
```

In the above example, the **GC** variable is assigned the number of graphics on the current page. The second statement selects the last created graphic on the current page.

---

`{button ,AL('VENTURA_GraphicFirst_Menu;vent_graphic;;;;',0,"Defaultoverview"),}` [Related Topics](#)

## .GraphicNext (VENTURA)

**.GraphicNext** .IgnoreGroup=*Boolean*, .Extend=*Boolean*

This command selects the next graphic ordered after the active graphic on the current page. This command is only valid when a graphic is currently selected.

Syntax	Description
.IgnoreGroup	If the next graphic is part of a group of graphics, specifies whether to select the individual graphic or the graphic group. Set to TRUE (-1) to select the individual graphic; otherwise, set to FALSE (0) to select the entire group which is the default if omitted.
.Extend	If graphics are currently selected, specifies whether to include the next graphic in the selection or to deselect the currently selected graphics. Set to TRUE (-1) to include. Set to FALSE (0) to deselect the currently selected graphics. FALSE is the default if this parameter is omitted.

### Note



This command cannot be recorded.



Graphics are ordered based on the order they are inserted on a page. This command is ignored if the last graphic on the page is selected when the command is issued.



You can use the [.GraphicPrev](#) command to select the previous graphic on the current page.



The **.IgnoreGroup** and **.Extend** parameters cannot both be set TRUE (-1).

### Example

```
.GraphicFirst  
.GraphicNext , TRUE
```

In the above example, the first line selects the first graphic on the current page and the second line selects the second graphic on the page, extending the selection.

---

{button ,AL("VENTURA\_GraphicNext\_Menu;vent\_graphic;;;','0,"Defaultoverview"),} [Related Topics](#)

## .GraphicPosition (VENTURA)

**.GraphicPosition** .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long*

This command changes the dimensions and position of a selected graphic.

Syntax	Description
.Left	Specifies the distance from the left side of the selected graphic's bounding box to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the left side is to the left of the base page frame. If this parameter is omitted, this setting is not changed.
.Top	Specifies the distance from the top of the selected graphic's bounding box to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the top side is above the top of the base page frame. If this parameter is omitted, this setting is not changed.
.Width	Specifies the width (the distance from the left side of the bounding box to right side) of the selected graphic's bounding box, in tenths of micron. If this parameter is omitted, this setting is not changed.
.Height	Specifies the height (the distance from the top of the bounding box to the bottom) of the selected graphic's bounding box, in tenths of micron. If this parameter is omitted, this setting is not changed.

### Note



The .ResizeObject command can also be used to resize a graphic.



The .MoveObject command can also be used to move a graphic.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.EditDuplicate
.GraphicPosition 4*M_INCH, 2.5*M_INCH, 3.75*M_INCH, 4.9*M_INCH
```

The above example draws an ellipse and duplicates it. The duplicate ellipse, still selected is then resized and moved.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_GraphicPosition\_Menu;vent\_graphic;;;',0,"Defaultoverview"),} Related Topics



# .GraphicPositionGet (VENTURA)

**.GraphicPositionGet** .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long*

This function returns the dimensions and coordinates of a selected graphic.

Syntax	Description
.Left	Specifies the numeric variable that is assigned the distance from the left side of the selected graphic's bounding box to the left side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the left side is to the left of the base page frame.
.Top	Specifies the numeric variable that is assigned the distance from the top of the selected graphic's bounding box to the top side of the <u>base page frame</u> , in tenths of micron. Negative values indicate the top side is above the top of the base page frame.
.Width	Specifies the numeric variable that is assigned the width of the selected graphic's bounding box, in tenths of micron.
.Height	Specifies the numeric variable that is assigned the height of the selected graphic's bounding box, in tenths of micron.

**Note**

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  If more than one graphic is selected, this function returns the attributes of the first selected graphic.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.EditDuplicate
.GraphicPositionGet LeftEllipse&, TopEllipse&, WidthEllipse&, HeightEllipse&
```

The above example draws an ellipse and duplicates it. The dimensions and coordinates of the duplicate ellipse are assigned to the **LeftEllipse**, **TopEllipse**, **WidthEllipse**, and **HeightEllipse** variables.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.







# .GraphicPrev (VENTURA)

**.GraphicPrev** .IgnoreGroup=*Boolean*, .IgnoreGroup=*Boolean*

This command selects the previous graphic ordered before the active graphic on the current page. This command is only valid when a graphic is currently selected.

Syntax	Description
.IgnoreGroup	If the previous graphic is part of a group of graphics, specifies whether to select the individual graphic or the graphic group. Set to TRUE (-1) to select the individual graphic; otherwise, set to FALSE (0) to select the entire group which is the default if omitted.
.Extend	If graphics are currently selected, specifies whether to include the previous graphic in the selection or to deselect the currently selected graphics. Set to TRUE (-1) to include. Set to FALSE (0) to deselect the currently selected graphics. FALSE is the default if this parameter is omitted.

## Note

-  This command cannot be recorded.
-  Graphics are ordered based on the order they are inserted on a page. This command is ignored if the last graphic on the page is selected when the command is issued.
-  You can use the [.GraphicNext](#) command to select the next graphic on the current page.
-  The **.IgnoreGroup** and **.Extend** parameters cannot both be set TRUE (-1).

## Example

`.GraphicPrev`

---

`{button ,AL(`vent_graphic;;;',0,"Defaultoverview"),}` [Related Topics](#)




# .GraphicRotate (VENTURA)

**.GraphicRotate .Angle=*long*, .CenterX=*long*, .CenterY=*long***

This command rotates a selected graphic.

Syntax	Description
.Angle	Specifies the angle of rotation in tenths of degrees (45 degrees equals 450 tenths of a degree), relative to the center of the graphic. Positive numbers result in counter-clockwise rotation, negative numbers result in clockwise rotation.
.CenterX	Specifies the horizontal center of rotation of the selected object. This is the distance from the left side <u>base page frame</u> to the horizontal center of rotation position.
.CenterY	Specifies the vertical center of rotation of the selected object. This is the distance from the left side <u>base page frame</u> to the vertical center of rotation position.

**Note**

-  If more than one graphic is selected, this command rotates first selected graphic.
-  The .RotateObject command can also be used to rotate a graphic.
-  This command cannot be recorded.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.GraphicRotate 450, 4*M_INCH, 5.5*M_INCH
```

The above example draws a rectangle 3 by 4 inches. The rectangle is then rotated 45 degrees from a specified center of rotation position.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

## .GraphicSkew (VENTURA)

**.GraphicSkew** .AngleX=*long*, .AngleY=*long*, .X=*long*, .Y=*long*

This command skews a selected graphic.

Syntax	Description
.AngleX	Specifies the angle to skew horizontally, in tenths of degrees (45 degrees equals 450 tenths of a degree), relative to the center of the graphic. Positive numbers result in counter-clockwise skew, negative numbers result in clockwise skew. If not specified, set to 0.
.AngleY	Specifies the angle to skew vertically, in tenths of degrees (45 degrees equals 450 tenths of a degree), relative to the center of the graphic. Positive numbers result in counter-clockwise skew, negative numbers result in clockwise skew. If not specified, set to 0.
.X	Specifies the horizontal center of skew of the selected object. This is the distance from the left side <u>base page frame</u> to the horizontal center of skew position. If omitted, the graphic's horizontal center position is used.
.Y	Specifies the vertical center of skew of the selected object. This is the distance from the left side <u>base page frame</u> to the vertical center of skew position. If omitted, the graphic's vertical center position is used.

### Note



If more than one graphic is selected, this command rotates first selected graphic.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.GraphicSkew 150, 100, 4*M_INCH, 5.5*M_INCH
```

The above example draws a rectangle 3 by 4 inches. The rectangle is then skewed horizontally 15 degrees and vertically 10 degrees from a specified center of skew position.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_GraphicSkew\_Menu;vent\_graphic;;;',0,"Defaultoverview"),} Related Topics

## .CrossReferenceGet (VENTURA)

**.CrossReferenceGet** .ReferenceType=*long*, .ReferToMarker=*string*, .ReferenceAll=*Boolean*, .Delimiter=*long*, .Format=*long*, .Uppercase=*Boolean*, .Capitalize=*Boolean*, .Insert1=*Boolean*, .Insert2=*Boolean*

This function returns the settings of a cross reference item at the text caret position.

Syntax	Description																										
.ReferenceType	<p>Specifies a numeric variable which is assigned the reference type:</p> <table><tr><td>0</td><td>Page Number</td></tr><tr><td>1</td><td>Chapter Number</td></tr><tr><td>2</td><td>Figure Number</td></tr><tr><td>3</td><td>Table Number</td></tr><tr><td>4</td><td>Section Number</td></tr><tr><td>5</td><td>Caption Text</td></tr><tr><td>6</td><td>Variable</td></tr><tr><td>7</td><td>First Match</td></tr><tr><td>8</td><td>Last Match</td></tr><tr><td>9</td><td>Current Match</td></tr><tr><td>10</td><td>Chapter and Page Number</td></tr><tr><td>11</td><td>Chapter and Section Number</td></tr><tr><td>12</td><td>Section and Page number</td></tr></table> <p>This parameter was updated in Corel VENTURA 8.</p>	0	Page Number	1	Chapter Number	2	Figure Number	3	Table Number	4	Section Number	5	Caption Text	6	Variable	7	First Match	8	Last Match	9	Current Match	10	Chapter and Page Number	11	Chapter and Section Number	12	Section and Page number
0	Page Number																										
1	Chapter Number																										
2	Figure Number																										
3	Table Number																										
4	Section Number																										
5	Caption Text																										
6	Variable																										
7	First Match																										
8	Last Match																										
9	Current Match																										
10	Chapter and Page Number																										
11	Chapter and Section Number																										
12	Section and Page number																										
.ReferToMarker	Specifies a string variable which is assigned the name of the marker to refer to.																										
.ReferenceAll	Specifies a numeric variable which is assigned the reference setting: TRUE (-1) to reference all occurrences throughout the publication; otherwise, FALSE (0).																										
.Delimiter	<p>Specifies a numeric variable which is assigned a value corresponding to how multiple instances of the selected marker are rendered</p> <table><tr><td>0</td><td>,,, ("See pages 1, 2, 3, 4")</td></tr><tr><td>1</td><td>,,,and ("See pages 1, 2, 3, and 4")</td></tr><tr><td>2</td><td>,,, or ("See pages 1, 2, 3, or 4")</td></tr><tr><td>3</td><td>x to y ("See pages 1 to 4")</td></tr><tr><td>4</td><td>x - y ("See pages 1 - 4")</td></tr></table>	0	,,, ("See pages 1, 2, 3, 4")	1	,,,and ("See pages 1, 2, 3, and 4")	2	,,, or ("See pages 1, 2, 3, or 4")	3	x to y ("See pages 1 to 4")	4	x - y ("See pages 1 - 4")																
0	,,, ("See pages 1, 2, 3, 4")																										
1	,,,and ("See pages 1, 2, 3, and 4")																										
2	,,, or ("See pages 1, 2, 3, or 4")																										
3	x to y ("See pages 1 to 4")																										
4	x - y ("See pages 1 - 4")																										
.Format	<p>Specifies a numeric variable which is assigned a value corresponding to the printed format of the cross reference:</p> <table><tr><td>0</td><td>Default</td></tr><tr><td>1</td><td>Text</td></tr><tr><td>2</td><td>Alphabetic</td></tr><tr><td>3</td><td>Roman Numerals</td></tr></table> <p>This parameter is ignored if <b>.ReferenceType</b> is set to 5 (Caption Text).</p>	0	Default	1	Text	2	Alphabetic	3	Roman Numerals																		
0	Default																										
1	Text																										
2	Alphabetic																										
3	Roman Numerals																										
.Uppercase	<p>Specifies a numeric variable which is assigned the uppercase setting: TRUE (-1) if the cross reference and any associated prefix are rendered using capital letters; otherwise, FALSE (0).</p> <p>This parameter is ignored if <b>.ReferenceType</b> is set to 5 (Caption Text).</p>																										
.Capitalize	<p>Specifies a numeric variable which is assigned the initial capital setting: TRUE (-1) if the cross reference and any associated prefix are rendered using initial capitals; otherwise, FALSE (0).</p> <p>This parameter is ignored if <b>.ReferenceType</b> is set to 5 (Caption Text).</p>																										
.Insert1	Specifies a numeric variable which is assigned a value corresponding to whether to insert the word "Chapter" when <b>.ReferenceType</b> is set to Chapter Number. TRUE (-1) if enabled; otherwise, FALSE (0).																										
.Insert2	Specifies a numeric variable which is assigned a value corresponding to whether to insert the word "Page" when <b>.ReferenceType</b> is set to Page Number. TRUE (-1) if enabled; otherwise, FALSE (0).																										

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This function cannot be recorded.



This function was introduced in Corel VENTURA 8.

### Example

```
DIM ra AS BOOLEAN
.CrossReferenceGet RT&, RTOM$, ra
```

The above example returns the first three parameters of this function to three variables.


---

`{button ,AL('VENT_insert;;;','0','Defaultoverview',)}` [Related Topics](#)

## .DateTimeGet (VENTURA)

**.DateTimeGet** .Format=*string*, .Language=*long*

This function returns the attributes of a date and time item at the text caret position.

Syntax	Description																		
.Format	Specifies a string variable which is assigned a code representing the format of the selected date and time item. For a listing of the date and time formatting codes, click  .																		
.Language	Specifies a numeric variable which is assigned the language for the selected date and time item: <table><tr><td>0</td><td>English</td></tr><tr><td>1</td><td>French</td></tr><tr><td>2</td><td>German</td></tr><tr><td>3</td><td>Italian</td></tr><tr><td>4</td><td>Spanish</td></tr><tr><td>5</td><td>Dutch</td></tr><tr><td>6</td><td>Portuguese</td></tr><tr><td>7</td><td>Swedish</td></tr><tr><td>8</td><td>Danish</td></tr></table>	0	English	1	French	2	German	3	Italian	4	Spanish	5	Dutch	6	Portuguese	7	Swedish	8	Danish
0	English																		
1	French																		
2	German																		
3	Italian																		
4	Spanish																		
5	Dutch																		
6	Portuguese																		
7	Swedish																		
8	Danish																		

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This function cannot be recorded.



This function was introduced in Corel VENTURA 8.

### Example

```
.DateTimeGet DTFormat$, DTLang&
```

The above example assigns the values of a selected date and time item to the variables **DTFormat** and **DTLang**.

---

{button ,AL('VENT\_insert;;;','0',"Defaultoverview",)} [Related Topics](#)

## .HiddenTextGet (VENTURA)

**.HiddenTextGet** *.HiddenText=string*

This function returns the hidden text at the text caret position.

Syntax	Description
<b>.HiddenText</b>	Specifies the string variable which is assigned the hidden text.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This function cannot be recorded.



This function was introduced in Corel VENTURA 8.

### Example

```
.HiddenTextGet HT$  
MESSAGE HT$
```

In the above example the hidden text at the text caret position is assigned to the **HT** string variable. The **HT** variable contents is then displayed in a message box.

---

{button ,AL("VENT\_insert;;;",0,"Defaultoverview",)} Related Topics

## .InsertCrossReference (VENTURA)

**.InsertCrossReference** **.ReferenceType=long**, **.ReferToMarker=string**, **.ReferenceAll=Boolean**, **.Delimiter=long**, **.Format=long**, **.Uppercase=Boolean**, **.Capitalize=Boolean**, **.Insert1=Boolean**, **.Insert2=Boolean**

This command inserts cross-references into your publication. Use the **.InsertMarker** command to insert cross-reference markers.

Syntax	Description
<b>.ReferenceType</b>	Specifies the type of reference to be inserted: 0 Page Number 1 Chapter Number 2 Figure Number 3 Table Number 4 Section Number 5 Caption Text 6 Variable 7 First Match 8 Last Match 9 Current Match 10 Chapter and Page Number 11 Chapter and Section Number 12 Section and Page number  This parameter was updated in Corel VENTURA 8.
<b>.ReferToMarker</b>	Specifies the name of the marker to refer to. The marker must already exist in the publication.
<b>.ReferenceAll</b>	Specifies whether to reference all occurrences throughout the publication. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
<b>.Delimiter</b>	Specifies how multiple instances of the selected marker will be rendered 0 ,,, ("See pages 1, 2, 3, 4") 1 ,,,and ("See pages 1, 2, 3, and 4") 2 ,,, or ("See pages 1, 2, 3, or 4") 3 x to y ("See pages 1 to 4") 4 x - y ("See pages 1 - 4")
<b>.Format</b>	Specifies the printed format of the cross reference: 0 Default 1 Text 2 Alphabetic 3 Roman Numerals  This parameter is ignored if <b>.ReferenceType</b> is set to 5 (Caption Text).
<b>.Uppercase</b>	Specifies whether the cross reference and any associated prefix will be rendered using capital letters. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). This parameter is ignored if <b>.ReferenceType</b> is set to 5 (Caption Text).
<b>.Capitalize</b>	Specifies whether the cross reference and any associated prefix will be rendered with the first letter of each word capitalized. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). This parameter is ignored if <b>.ReferenceType</b> is set to 5 (Caption Text).
<b>.Insert1</b>	Specifies whether to insert the word "Chapter" when <b>.ReferenceType</b> is set to Chapter Number. Set to TRUE (-1) to enable this option; otherwise, FALSE (0) which is the default if omitted.
<b>.Insert2</b>	Specifies whether to insert the word "Page" when <b>.ReferenceType</b> is set to Page Number. Set to TRUE (-1) to enable this option; otherwise, FALSE (0) which is the default if omitted.

### Note

**I** This command corresponds to the Insert Number/Cross-reference dialog box in Corel VENTURA. Click Insert, Number/Cross-Reference.

### Example

```
.InsertCrossReference 0, "Author", FALSE, 0, 2, TRUE,, TRUE
```

The above example inserts a page-number cross-reference to the Author maker. Uppercase alphabetic formatting is used with the cross-reference, and the word PAGE is inserted before the page number.

---


{button ,AL('VENT\_insert;;;',0,"Defaultoverview"),} [Related Topics](#)



## .InsertDateTime (VENTURA)

**.InsertDateTime** **.Format=string**, **.Language=long**, **.Update=Boolean**

This command inserts the date, time, or both in the format you specify.

Syntax	Description																		
<b>.Format</b>	Specifies a code representing the format of the date, time, or both. For a listing of the date and time formatting codes, click  .																		
<b>.Language</b>	Specifies the language for the time and date: <table><tr><td>0</td><td>English</td></tr><tr><td>1</td><td>French</td></tr><tr><td>2</td><td>German</td></tr><tr><td>3</td><td>Italian</td></tr><tr><td>4</td><td>Spanish</td></tr><tr><td>5</td><td>Dutch</td></tr><tr><td>6</td><td>Portuguese</td></tr><tr><td>7</td><td>Swedish</td></tr><tr><td>8</td><td>Danish</td></tr></table> <p>If this parameter is not specified, system setting are used.</p>	0	English	1	French	2	German	3	Italian	4	Spanish	5	Dutch	6	Portuguese	7	Swedish	8	Danish
0	English																		
1	French																		
2	German																		
3	Italian																		
4	Spanish																		
5	Dutch																		
6	Portuguese																		
7	Swedish																		
8	Danish																		
<b>.Update</b>	Specifies whether the date and time is updated each time the screen refreshes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).																		

### Note



This command corresponds to the Date and Time dialog box in Corel VENTURA. Click Insert, Date and Time.

### Example

```
.InsertDateTime
```

The above example inserts the date and time with the Windows default regional settings.

```
.InsertDateTime "MMMM d, yyyy"
```

The above example inserts a date similar to the following: January 14, 1996.

---

**{button ,AL("VENT\_insert;;;",0,"Defaultoverview",)} Related Topics**

## **.InsertEndnote (VENTURA)**

### **.InsertEndnote**

This command inserts an endnote in the active publication at the insertion point.

#### **Note**



This command corresponds to the Endnote command on the Insert menu in Corel VENTURA. Click Insert, Endnote.

#### **Example**

```
.FileNew      'create a new publication
.PageFirstLine 'place insertion point in base frame
.TypeText "This is text in the base page frame" 'type some text
.InsertEndnote 'insert an endnote and place insertion point in endnote frame
.TypeText "This is endnote text" 'type some endnote text
```

For an explanation of the example, see the script comments (text that follows the apostrophes).


---

{button ,AL('VENT\_insert;;;','0,"Defaultoverview",)} [Related Topics](#)


# .InsertEquation (VENTURA)

**.InsertEquation** .Equation=*string*

This command inserts an equation at the insertion point or opens the Insert Equation dialog box in Corel VENTURA.

Syntax	Description
.Equation	Specifies the equation to insert using VENTURA's EQN language. To create the equation $\cos(\theta_1) + \sin(\theta_2)$ use the following string: cos ( theta sub 1 ) + sin ( theta sub 2 ) To learn more about VENTURA's EQN language, click  . If this parameter is omitted, the Insert Equation dialog box opens.

**Note**

 This command corresponds to the Insert Equation dialog box in Corel VENTURA. Click Insert, Equation.

**Example**

`.InsertEquation`

The above example opens the Insert Equation dialog box.

`.InsertEquation "cos ( theta sub 1 )+ sin ( theta sub 2 )"`

The above example inserts the following equation at the insertion point:

$\cos(\theta_1) + \sin(\theta_2)$

## **.InsertFootnote (VENTURA)**

### **.InsertFootnote**

This command inserts a footnote in the active publication at the insertion point.

#### **Note**



This command corresponds to the Footnote command on the Insert menu in Corel VENTURA. Click Insert, Footnote.

#### **Example**

```
.FileNew      'create a new publication
.PageFirstLine 'place insertion point in base frame
.TypeText "This is text in the base page frame" 'type some text
.InsertFootnote 'insert a footnote and place insertion point in footnote frame
.TypeText "This is footnote text" 'type some footnote text
.PageLastLine  'send insertion point back to last line in the base frame
.TextEndOfLine 'send insertion point to end of line, after the footnote
```

For an explanation of the example, see the script comments (text that follows the apostrophes).

---

{button ,AL('VENT\_insert;;;','0',"Defaultoverview",)} [Related Topics](#)

## **.InsertHiddenText (VENTURA)**

**.InsertHiddenText .HiddenText=string**

This command inserts hidden text at the insertion point.

Syntax	Description
<b>.HiddenText</b>	Specifies the hidden text to insert.

### **Note**



Corel VENTURA automatically disables Flow Text Around Frame for frames anchored using the At Anchor position option.

### **Example**

```
.InsertHiddenText "You can't see me"
```

The above example insert the hidden text, **You can't see me** at the insertion point.

---

`{button ,AL("VENT_insert;;;",0,"Defaultoverview",)}` [Related Topics](#)

## .InsertHyperlink (VENTURA)

**.InsertHyperlink** .LinkType=*long*, .DisplayText=*string*, .URL=*string*, .Marker=*string*, .TargetFrame=*string*,

This command inserts a hyperlink in the text at the insertion point.

Syntax	Description
.LinkType	Specifies the type of the hyperlink: 0 Internet URL 1 Email address 2 File 3 Chapter
.DisplayText	Specifies the text to be displayed for the hyperlink.
.URL	Specifies the Internet URL, email address, file name, or chapter name. This parameter depends on the setting of .LinkType.
.Marker	If .LinkType is set to Chapter, the specifies the marker.
.TargetFrame	If .LinkType is set to 0, 2, or 3, the specifies the target frame. For information about target frames, see "Creating links in an HTML document" in the Corel VENTURA 8 online Help. If omitted, set to Default.

### Note



This command corresponds to clicking Insert, Hyperlink in Corel VENTURA.

### Example

```
.InsertHyperlink 0, "Visit Corel", "http://www.corel.com"  
.InsertHyperlink 1, "Send me an email", "myname@server.com"
```



In the above example, the first line creates a hyperlink for a Internet URL and the second line creates a hyperlink for an email address.

---

{button ,AL("VENT\_insert;;;",0,"Defaultoverview"),} [Related Topics](#)

## **.InsertMarker (VENTURA)**

**.InsertMarker .MarkerName=string**

This command inserts a cross reference marker  the place that the cross reference refers to  at the insertion point. To insert the cross reference, use the [.InsertCrossReference](#) command.

---


### **Syntax**

### **Description**

**.MarkerName**

Specifies the name of new cross reference marker or re-specifies the name of an existing marker at the insertion point.

### **Note**

 This command corresponds to the Marker command on the Insert menu in Corel VENTURA. Click Insert, Marker.

### **Example**

```
.InsertMarker "New Marker"
```

The above command inserts the marker New Marker.

---

{button ,AL("VENT\_insert;;;",0,"Defaultoverview",)} [Related Topics](#)

## .MarkerGet (VENTURA)

**.MarkerGet** *.MarkerName=string*

This function returns the marker at the text caret position.

Syntax	Description
<b>.MarkerName</b>	Specifies the string variable which is assigned the name of the marker.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This function cannot be recorded.



This function was introduced in Corel VENTURA 8.

### Example

```
.MarkerGet MN$
```

In the above example the marker's name at the text caret position is assigned to the **MN** string variable.

---


{button ,AL("VENT\_insert;;;",0,"Defaultoverview",)} Related Topics




## .InsertSymbol (VENTURA)

**.InsertSymbol** .FontName=*string*, .CharCode=*long*

This command inserts symbols and special characters at the insertion point from fonts on your system.

Syntax	Description																																						
.FontName	Specifies the name of the font to use. For a list of installed fonts, see the font box on the text toolbar.																																						
.CharCode	<p>Specifies the symbol or special character to insert. It is specified using ANSI character values of the current font. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters. For symbol fonts, you also see the Insert Character dialog box (click Insert, Insert Item, Character) for ANSI values  the values are listed at the bottom of the dialog box and are preceded by <b>ALT+0</b>.</p> <p>The following list notes some commonly used symbols and special characters, and their ANSI values. These numbers are dependent on the font being used but most standard character fonts use these values:</p> <table><tr><td>34</td><td>Straight double quote</td></tr><tr><td>145</td><td>Typographical open single quote</td></tr><tr><td>146</td><td>Typographical close single quote</td></tr><tr><td>147</td><td>Typographical open double quote</td></tr><tr><td>148</td><td>Typographical close double quote</td></tr><tr><td>150</td><td>En dash</td></tr><tr><td>151</td><td>Em dash</td></tr><tr><td>153</td><td>Trademark symbol</td></tr><tr><td>169</td><td>Copyright symbol</td></tr><tr><td>174</td><td>Registered symbol</td></tr><tr><td>183</td><td>Bullet symbol</td></tr></table> <p>The following list notes some VENTURA text formatting characters and their ANSI values:</p> <table><tr><td>9</td><td>Tab</td></tr><tr><td>10</td><td>Paragraph return</td></tr><tr><td>13</td><td>Forced line break</td></tr><tr><td>17</td><td>Em space</td></tr><tr><td>18</td><td>En space</td></tr><tr><td>19</td><td>Figure space</td></tr><tr><td>20</td><td>Thin space</td></tr><tr><td>21</td><td>Non-breaking space</td></tr></table>	34	Straight double quote	145	Typographical open single quote	146	Typographical close single quote	147	Typographical open double quote	148	Typographical close double quote	150	En dash	151	Em dash	153	Trademark symbol	169	Copyright symbol	174	Registered symbol	183	Bullet symbol	9	Tab	10	Paragraph return	13	Forced line break	17	Em space	18	En space	19	Figure space	20	Thin space	21	Non-breaking space
34	Straight double quote																																						
145	Typographical open single quote																																						
146	Typographical close single quote																																						
147	Typographical open double quote																																						
148	Typographical close double quote																																						
150	En dash																																						
151	Em dash																																						
153	Trademark symbol																																						
169	Copyright symbol																																						
174	Registered symbol																																						
183	Bullet symbol																																						
9	Tab																																						
10	Paragraph return																																						
13	Forced line break																																						
17	Em space																																						
18	En space																																						
19	Figure space																																						
20	Thin space																																						
21	Non-breaking space																																						

### Note

 This command corresponds to the Symbol command on the Insert menu in Corel VENTURA. Click Insert, Symbol.

### Example

```
.InsertSymbol "Courier", 174
```

The above example inserts the Registered symbol using the Courier font.

---

{button ,AL(^VENT\_insert;;;',0,"Defaultoverview",)} [Related Topics](#)

## .InsertURL (VENTURA)

**.InsertURL** *.URL=string, .DisplayText=string, .ClearFrames=Boolean*

This command inserts a Uniform Resource Locator (URL) at the insertion point. The URL appears as a hyperlink when the publication is converted to HTML or Barista and viewed in a browser.

Syntax	Description
<b>.URL</b>	Specifies the name and address of the information to link to from the HTML or Barista publication. Be sure to include the server type, for example, http:// or gopher://.
<b>.DisplayText</b>	Specifies the text to appear in the HTML or Barista publication. Readers viewing the publication in a browser can click the text to display the information at the specified URL address.
<b>.ClearFrames</b>	Specifies whether to have the information at the specified URL address replace the content in all browser frames or only in the browser's main frame, leaving the other frames alone. Set to TRUE (-1) to replace in all browser frames; otherwise, set to FALSE (0) which is the default if omitted.

### Note



This command corresponds to the Insert URL dialog box in Corel VENTURA. Click Insert, URL.

### Example

```
.InsertURL "http://www.corel.com/", "Corel's home page"
```

The above example inserts the URL that points to Corel Corporation's World Wide Web home page.

---

**{button ,AL("VENT\_insert;;;",0,"Defaultoverview"),}** Related Topics

## **.InsertVariable (VENTURA)**

**.InsertVariable** *.VariableName=string*

This command insert a variable marker at the insertion point. A variable marker is a placeholder for text or numbers used in cross references that change according to the variable's current definition. When you update the publication, VENTURA inserts the variable's updated information at all specified locations in the publication.

Syntax	Description
<b>.VariableName</b>	Specifies the name of a new variable marker at the insertion point.

### **Note**



This command corresponds to the Insert Variable Marker dialog box in Corel VENTURA. Click Insert, Variable.

### **Example**

```
.InsertVariable "Temp"
```

The above example inserts the variable Temp at the insertion point.

---

{button ,AL('VENT\_insert;;;','0',"Defaultoverview"),} [Related Topics](#)

## .IsFormatError (VENTURA)

**ReturnValue = .IsFormatError ( )**

This function returns a value indicating if there is a formatting error (red X appears in Corel VENTURA) in the current selection (for example, a frame or text).

Syntax	Description
<b>.ReturnValue</b>	Specifies a Boolean variable which is a value indicating whether there is a formatting error (red X appears in Corel VENTURA) in the current selection: TRUE (-1) if a formatting error exists; otherwise, FALSE (0). This variable must be <u>explicitly</u> declared.

### Note



This function cannot be recorded.



This function was introduced in Corel VENTURA 8.

### Example

```
DIM IsError AS BOOLEAN  
IsError = .IsFormatError ( )
```

In the above example the formatting error status of the current selection is assigned to the **IsError** Boolean variable.

---

{button ,AL("VENT\_insert;;;",0,"Defaultoverview",)} [Related Topics](#)

## .VariableGet (VENTURA)

**.VariableGet** *.VariableName=string*

This function returns the name of the Corel VENTURA variable at the text caret position.

Syntax	Description
<b>.VariableName</b>	Specifies the string variable which is assigned the name of the Corel VENTURA variable.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This function cannot be recorded.



This function was introduced in Corel VENTURA 8.

### Example

```
.VariableGet NV$
```

In the above example the name of the Corel VENTURA variable at the text caret position is assigned to the **VN** string variable.

---

{button ,AL("VENT\_insert;;;",0,"Defaultoverview",)} Related Topics

## **.ActivateLibrary (VENTURA)**

**.ActivateLibrary** *.FileName=string*

This command activates a VENTURA library (or opens it if not already open)

Syntax	Description
<b>.FileName</b>	Specifies the name and the path of the library to set active or open.

### **Note**

**I** This command cannot be recorded.

### **Example**

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryNew "C:\Core1\Ventura\Samples.vlb"  
.ActivateLibrary "C:\Win95\Desktop\MyLib.vlb"
```

The above example opens a VENTURA library named MYLIB.VLB, and creates the SAMPLES.VLB library. The MYLIB library is then activated.

---

{button ,AL("Ventura\_ActivateLibrary\_Menu;vent\_library;;;','0,"Defaultoverview",,)} Related Topics


## .CountLibraries (VENTURA)

ReturnValue = .CountLibraries ( )

This function returns the number of open VENTURA libraries.

Syntax	Description
ReturnValue	The numeric variable that is assigned the number of open VENTURA libraries.

### Note

 This command cannot be recorded.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryNew "C:\Corel\Ventura\Samples.vlb"  
LibraryNums& = .CountLibraries ( )
```

The above example opens a VENTURA library named MYLIB.VLB and creates the SAMPLES.VLB library. The open number of libraries is returned to the **LibraryNums** variable.

---

{button ,AL("Ventura\_CountLibraries\_Menu;vent\_library;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FileLibraryClose (VENTURA)

.FileLibraryClose .CloseAll=*Boolean*

This command closes open VENTURA libraries.

Syntax	Description
.CloseAll	Specifies whether to close all open libraries or the active library. Set to TRUE (-1) to close all libraries. Set to FALSE (0) to close the active library. If the parameter is omitted, the active library is closed.

### Note

 This command cannot be recorded.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\YourLib.vlb"  
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibrarySelectItem 3  
.FileLibraryCopy  
.FileLibraryClose -1
```

The above example opens the YourLib and MyLib VENTURA libraries. The third item in MyLib is selected. The third item is then copied from the VENTURA library to the Clipboard. The last statement then closes all the open libraries.

---

{button ,AL("VENTURA\_FileLibraryClose\_Menu;vent\_library;;;','0,"Defaultoverview",)} [Related Topics](#)





## .FileLibraryCopy (VENTURA)

### .FileLibraryCopy

This command copies the selected item from the active VENTURA library and places it on the Clipboard.

#### Note

 This command corresponds to the Copy command on the right mouse button menu in the VENTURA Library dialog box. Select an object in the dialog box and click the right mouse button. Click Copy.

 This command cannot be recorded.

#### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibrarySelectItem 3  
.FileLibraryCopy  
.FileLibraryNew "C:\Win95\Desktop\SpecialLib.vlb"  
.FileLibraryPaste
```

The above example opens the MYLIB library and selects the third item in it. The third item is then copied from the library and pasted into the newly created SPECIALLIB library.

---

{button ,AL('VENTURA\_FileLibraryCopy\_Menu;vent\_library;;;',0,"Defaultoverview"),} [Related Topics](#)


## .FileLibraryCreateItem (VENTURA)

**.FileLibraryCreateItem** *.FileName=string*

This command adds a specified item to the active VENTURA library.

Syntax	Description
<b>.FileName</b>	Specifies the name and path of the file to add to the active library.

### Note

 This command cannot be recorded.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryCreateItem "C:\MyFiles\Bruins.bmp"
```

The above example opens the MYLIB library and adds the BRUINS.BMP to it.

---

{button ,AL("VENTURA\_FileLibraryCreateItem\_Menu;vent\_library;;;','0,"Defaultoverview"),} [Related Topics](#)

## **.FileLibraryCut (VENTURA)**

### **.FileLibraryCut**

This command removes the selected item from the active VENTURA library and places it on the Clipboard.

#### **Note**



This command cannot be recorded.



This command corresponds to the Cut command on the right mouse button menu in the VENTURA Library dialog box. Select an object in the dialog box and click the right mouse button. Click Cut.



Use the **.FileLibraryDelete** command to remove the selection, and have it not placed on the Clipboard.

#### **Example**

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibrarySelectItem 3  
.FileLibraryCut  
.FileLibraryNew "C:\Win95\Desktop\SpecialLib.vlb"  
.FileLibraryPaste
```

The above example opens the MYLIB library and selects the third item in it. The third item is then cut from the library and pasted into the newly created SPECIALLIB library.

---

{button ,AL("VENTURA\_FileLibraryCut\_Menu;vent\_library;;;','0,"Defaultoverview"),} Related Topics

## **.FileLibraryDelete (VENTURA)**

### **.FileLibraryDelete**

This command removes the selected item from the active VENTURA library. The selection is not placed on the Clipboard. To place the removed selection on the Clipboard, use the [\*\*.FileLibraryCut\*\*](#) command instead.

#### **Note**



This command cannot be recorded.



This command corresponds to the Delete command on the right mouse button menu in the VENTURA Library dialog box. Select an object in the dialog box, and click the right mouse button. Click Delete.

#### **Example**

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibrarySelectItem 4  
.FileLibraryCut
```

The above example opens the MYLIB library and selects the fourth item in it. The fourth item is then removed from the library.

---

{button ,AL('VENTURA\_FileLibraryDelete\_Menu;vent\_library;;;',0,"Defaultoverview"),} [Related Topics](#)

## .FileLibraryFindItem (VENTURA)

**.FileLibraryFindItem** .ItemName=*string*

This command finds a specified item in the active VENTURA library and selects it.

Syntax	Description
.ItemName	Specifies the name of the item to find in the library. If this parameter is omitted, the command searches for the previously defined <b>ItemName</b> parameter. You can use wildcards in this parameter - see the second example below for more information.

### Note



This command cannot be recorded.



This command corresponds to the Find command on the right mouse button menu in the VENTURA Library dialog box. Click the right mouse button and click Find.



Searches begin from the first library item.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryFindItem "Wave Sound"  
.EditDelete  
.FileLibraryFindItem
```

The above example opens the MYLIB library and searches for the item named Wave Sound. The first Wave Sound item found is deleted. The last line searches for another item named Wave Sound.

```
.FileLibraryOpen "c:\corel\graphics8\ventura\library_v8\dividers_v8.vlb"  
FOR i%=1 TO .FileLibraryItemCount()  
    .FileLibraryFindItem "*.gif"  
    .FileLibraryDelete  
NEXT i%
```

The above example opens the dividers\_v8 VENTURA library and searches for all GIF files and deletes each found GIF file.

---

{button ,AL('VENTURA\_FileLibraryFindItem\_Menu;vent\_library;;;','0,"Defaultoverview",)} Related Topics

## **.FileLibraryItemCount (VENTURA)**

**ReturnValue** = .FileLibraryItemCount ( )

This function returns the number of items in the active VENTURA library.

Syntax	Description
<b>ReturnValue</b>	The numeric variable that is assigned the number of items in the active VENTURA library.

### **Note**

**I** This command cannot be recorded.

### **Example**

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
LibItems = .FileLibraryItemCount ( )
```

The above example opens a VENTURA library named MYLIB.VLB. The number of items in this library is returned to the **LibItems** variable.

---

{button ,AL('VENTURA\_FileLibraryItemCount\_Menu;vent\_library;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FileLibraryItemProperties (VENTURA)

**.FileLibraryItemProperties .Name=*string*, .Description=*string***

This command sets the properties of a selected item in the active VENTURA library.

Syntax	Description
<b>.Name</b>	Specifies the name of the selected library item.
<b>.Description</b>	Specifies the description of the selected library item.

### Note



This command cannot be recorded.



This command corresponds to the Properties command on the right mouse button menu in the VENTURA Library dialog box. Select an object in the dialog box and click the right mouse button. Click Properties.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibrarySelectItem 3  
.FileLibraryItemProperties "Mustang bitmap", "low resolution"
```

The above example opens the MYLIB library and selects the third item in it. The third item is renamed Mustang bitmap and is described as low resolution.

---

{button ,AL('VENTURA\_FileLibraryItemProperties\_Menu;vent\_library;;;',0,"Defaultoverview",)} Related Topics

## .FileLibraryItemPropertiesGet (VENTURA)

**.FileLibraryItemPropertiesGet** .Name=*string*, .Description=*string*, .Size=*long*, .Type=*long*, .SubType=*long*

This function returns the properties of the selected item in the active VENTURA library.

Syntax	Description
.Name	Specifies the string variable that is assigned the selected library item's name.
.Description	Specifies the string variable that is assigned the selected library item's description.
.Size	Specifies the numeric variable that is assigned the selected library item's size in bytes.
.Type	Specifies the numeric variable that is assigned the selected library item's type: 0 Unknown 1 Chapter component 2 File 3 Frame 4 Graphics 5 OLE object 6 Tag 8 Text
.SubType	This parameter was added in Core! VENTURA 8. If the library item type is a file or chapter, specifies the numeric variable that is assigned the selected library item's sub-type: 9 Page tag 10 Style sheet 11 Text file 12 Graphic file

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibrarySelectItem 3  
.FileLibraryItemPropertiesGet Name3$, Desc3$, Size3&, Type3&
```

The above example opens the MYLIB library and selects the third item in it. The third item's properties are then returned to the **Name3**, **Desc3**, **Size3**, and **Type3** variables.

---

{button ,AL("VENTURA\_FileLibraryItemPropertiesGet\_Menu;vent\_library;;;','0,"Defaultoverview",)} [Related Topics](#)







# .FileLibraryNew (VENTURA)

.FileLibraryNew .FileName=*string*

This command creates and opens a new VENTURA library. Libraries can contain items such as text, pictures, tags, and even entire chapters. You can move items from pages into an open library, from an open library onto pages, and between open libraries.

Syntax	Description
.FileName	Specifies the name and the path of the library to create.

Note

-  This command cannot be recorded.
-  The specified library path must exist or an error occurs.
-  The **.FileLibraryNew** command will overwrite an existing library with the same name.
-  This command corresponds to the New command on the VENTURA Library flyout of the Tools menu. Click Tools, VENTURA Library, New.

Example

```
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"
```

The above example creates a VENTURA library named MYLIB.VLB.

---

{button ,AL('VENTURA\_FileLibraryNew\_Menu;vent\_library;;;','0,"Defaultoverview",')} [Related Topics](#)




# .FileLibraryOpen (VENTURA)

**.FileLibraryOpen** *.FileName=string*

This command opens a previously saved Corel VENTURA Library. Libraries can contain items such as text, pictures, tags and even entire chapters. You can move items from pages into an open library, from an open library onto pages, and between open libraries.

Syntax	Description
<b>.FileName</b>	Specifies the name and the path of the library to open.

**Note**

-  This command cannot be recorded.
-  This command corresponds to the Open command on the VENTURA Library flyout of the Tools menu. Click Tools, VENTURA Library, Open.
-  A library opened with the **.FileOpenLibrary** command becomes the active library.

**Example**

`.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"`

The above example opens a VENTURA library named MYLIB.VLB.

---

`{button ,AL('VENTURA_FileLibraryOpen_Menu;vent_library;;;',0,"Defaultoverview"),}` [Related Topics](#)

## **.FileLibraryPaste (VENTURA)**

### **.FileLibraryPaste**

This command places a copy of the Clipboard contents into the active VENTURA library. The pasted item becomes the last item in the active library unless an item is already selected in the active library. If an item is already selected in the library, the pasted item is placed before it in the library.

#### **Note**



This command cannot be recorded.



This command corresponds to the Paste command on the right mouse button menu in the VENTURA Library dialog box. Click the right mouse button and click Copy.



The original cut or copied object remains on the Clipboard until you copy or cut another object, or end the current Windows session.



You can paste VENTURA tags into a library by first copying them with the following commands:

**.BorderTagCopy**

**.CharacterTagCopy**

**.FrameTagCopy**

**.ParaTagCopy**

#### **Example**

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibrarySelectItem 3  
.FileLibraryCut  
.FileLibraryNew "C:\Win95\Desktop\SpecialLib.vlb"  
.FileLibraryPaste
```

The above example opens the MYLIB library and selects the third item in it. The third item is then cut from the library and pasted into the newly created SPECIALLIB library.

---

{button ,AL('VENTURA\_FileLibraryPaste\_Menu;vent\_library;;;','0,"Defaultoverview",')} [Related Topics](#)

## .FileLibraryPropertiesGet (VENTURA)

**.FileLibraryPropertiesGet** .FileName=*string*, .TotalItems=*long*, .CreatedOn=*string*, .ModifiedOn=*string*,

This function returns the properties of the active Corel VENTURA library.

Syntax	Description
.FileName	Specifies the string variable that is assigned the selected library's name and path.
.TotalItems	Specifies the numeric variable that is assigned the number of items in the library.
.CreatedOn	Specifies the string variable that is assigned the creation date of the library.
.ModifiedOn	Specifies the string variable that is assigned the date that the library was last changed.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib1.vlb"  
.FileLibraryOpen "C:\Win95\Desktop\MyLib2.vlb"  
.FileLibraryOpen "C:\Win95\Desktop\MyLib3.vlb"  
.FileLibraryPropertiesGet LibName$, TI&, CO$, MO$
```

The above example opens three libraries and returns the properties of the active library (MYLIB3.VLB) to the **LibName**, **TI**, **CO**, **MO** variables.

---

{button ,AL('VENTURA\_FileLibraryPropertiesGet\_Menu;vent\_library;;;',0,"Defaultoverview"),} Related Topics

## **.FileLibrarySave (VENTURA)**

### **.FileLibrarySave**

This command saves changes made to the active VENTURA Library.

#### **Note**



This command cannot be recorded.

#### **Example**

```
.ParagraphTagCopy "TradeMark"  
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste  
.FileLibrarySave
```

The above example copies the paragraph tag TradeMark, creates a VENTURA library named MYLIB.VLB, and pastes the TradeMark tag into the new library. The library MYLIB.VLB is then saved.

---

{button ,AL('VENTURA\_FileLibraryPropertiesGet\_Menu;vent\_library;;;','0,"Defaultoverview",')} [Related Topics](#)

## .FileLibrarySelectItem (VENTURA)

**.FileLibrarySelectItem** .ItemIndex=*long*

This command selects a specified item in the active VENTURA library.

Syntax	Description
.ItemIndex	Specifies an item's index number. The first library item is 1, the second item is 2, and so on. If this parameter is omitted or invalid, the first item is selected.

### Note



This command cannot be recorded.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibrarySelectItem 3
```

The above example opens the MYLIB library and selects the third item in it.

---

{button ,AL('VENTURA\_FileLibrarySelectItem\_Menu;vent\_library;;;','0,"Defaultoverview",')} [Related Topics](#)

## .FileLibraryView (VENTURA)

**.FileLibraryView .Largelcons=*Boolean*, .Details=*Boolean***

This command sets the view options for the active VENTURA library.

Syntax	Description
<b>.Largelcons</b>	Specifies whether to display items with large or small icons in the active VENTURA library. Set to TRUE (-1) to show large icons; otherwise, set to FALSE (0).
<b>.Details</b>	Specifies whether to display details for the items in the active VENTURA library. Set to TRUE (-1) to display details; otherwise, set to FALSE (0).

### Note



This command cannot be recorded.



This command corresponds to the View command on the right mouse button menu in the VENTURA Library dialog box. Click the right mouse button and click View.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryView -1 , 0
```

The above example opens the MYLIB library. The view options for MYLIB are set to large icons without details.

---

**{button ,AL('VENTURA\_FileLibraryView\_Menu;vent\_library;;;',0,"Defaultoverview"),}** [Related Topics](#)

## .FileLibraryViewFilter (VENTURA)

**.FileLibraryViewFilter .ChpComponents=Boolean, .Files=Boolean, .Frames=Boolean, .Graphics=Boolean, .OLEObjects=Boolean, .Tags=Boolean, .Tables=Boolean, .Text=Boolean**

This command sets the view filter options for the active VENTURA library.

Syntax	Description
<b>.ChpComponents</b>	Specifies whether to show or to hide chapter components in the active VENTURA library. Set to TRUE (-1) to show chapter components; otherwise, set to FALSE (0).
<b>.Files</b>	Specifies whether to show or to hide files in the active VENTURA library. Set to TRUE (-1) to show chapter components; otherwise, set to FALSE (0).
<b>.Frames</b>	Specifies whether to show or to hide frames in the active VENTURA library. Set to TRUE (-1) to show chapter components; otherwise, set to FALSE (0).
<b>.Graphics</b>	Specifies whether to show or to hide graphics in the active VENTURA library. Set to TRUE (-1) to show chapter components; otherwise, set to FALSE (0).
<b>.OLEObjects</b>	Specifies whether to show or to hide OLE objects in the active VENTURA library. Set to TRUE (-1) to show chapter components; otherwise, set to FALSE (0).
<b>.Tags</b>	Specifies whether to show or to hide tags in the active VENTURA library. Set to TRUE (-1) to show chapter components; otherwise, set to FALSE (0).
<b>.Tables</b>	Specifies whether to show or to hide tables in the active VENTURA library. Set to TRUE (-1) to show chapter components; otherwise, set to FALSE (0). This parameter is not valid in Corel VENTURA 8 since tables are treated as text.
<b>.Text</b>	Specifies whether to show or to hide text in the active VENTURA library. Set to TRUE (-1) to show chapter components; otherwise, set to FALSE (0).

### Note



This command cannot be recorded.



This command corresponds to the Filter command on the right mouse button menu in the VENTURA Library dialog box. Click the right mouse button, and click Filter.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryViewFilter -1 , 0, 0 , 0, -1, 0, 0, 0
```

The above example opens the MYLIB library. The view filter options are set so that only chapter components and OLE objects are viewed in MYLIB.

---

{button ,AL('VENTURA\_FileLibraryViewFilter\_Menu;vent\_library;;;',0,"Defaultoverview",)} [Related Topics](#)



## .FileLibraryViewSort (VENTURA)

**.FileLibraryViewSort .SortType=*long***

This command sets the sort option for the active VENTURA library.

Syntax	Description
<b>.SortType</b>	Specifies the sort option for the active VENTURA library: 1      by name 2      by type 3      by date 4      by size

### Note



This command cannot be recorded.



This command corresponds to the Sort command on the right mouse button menu in the VENTURA Library dialog box. Click the right mouse button and click Sort.

### Example

```
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryViewSort 4
```

The above example opens the MYLIB library . The items in the library are set to be sorted by size.

---

{button ,AL("VENTURA\_FileLibraryViewSort\_Menu;vent\_library;vent\_library;;;",0,"Defaultoverview",)} Related Topics

## **.CurrentMasterPage (VENTURA)**

**ReturnString\$ = .CurrentMasterPage ( )**

This function returns the name of the active Page tag (Master Page).

<b>Syntax</b>	<b>Description</b>
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of the active Page tag (.

### **Example**

```
CurMP$ = .CurrentMasterPage ( )
```

The **CurMP** variable is assigned the name of the active Page tag (Master Page).

---


**{button ,AL("VENTURA\_MasterPage;;;','0,"Defaultoverview",)} Related Topics**

# .FormatMasterPage (VENTURA)


**.FormatMasterPage** .Portrait=*Boolean*, .PaperType=*long*, .Width=*long*, .Height=*long*, .MasterPage=*string*

This command is obsolete in Corel VENTURA 8. It has been replaced by [.FormatPageLayout](#).

This command sets the orientation, size, and layout style of the active or specified Master Page.

Syntax	Description
.Portrait	Specifies the page orientation. Set to TRUE (-1) for portrait. Set to FALSE (0) for landscape. If omitted, the default is TRUE.
.PaperType	Specifies the paper type. The following is a list of some common paper types: 0 Letter 1 Legal 2 Tabloid 34 Custom  Click  for a complete list of paper types.
.Width	Required parameter if <b>.PaperType</b> is set to 34 (Custom). Specifies the paper width in tenths of a micron.
.Height	Required parameter if <b>.PaperType</b> is set to 34 (Custom). Specifies the paper height in tenths of a micron.
.MasterPage	Specifies the Master page to apply the changes to. If not specified, the active Master page is used.

## Note

 This command corresponds to the Master Page Properties dialog box in Corel VENTURA. Click Page, Master Page Properties.

## Example

`.FormatMasterPage TRUE, 0`

The above example sets the active Page tag to a portrait orientation using a letter size paper type.

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatMasterPage TRUE, 34, 5*M_INCH, 7*M_INCH
```

The above example sets the active Page tag to a portrait orientation using a paper size of 5 by 7 inches. The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

`{button ,AL('VENTURA_MasterPage;;;','0,"Defaultoverview",)} Related Topics`

# .FormatMasterPageApply (VENTURA)


**.FormatMasterPageApply** **.MasterPage=string**, **.FromPage=long**, **.ToPage=long**, **.LeftPages=Boolean**, **.RightPages=Boolean**, **.OldMasterPage=string**

This command is obsolete in Corel VENTURA 8. It has been replaced by [.PageTagApply](#).

This command changes the Master Page on specified pages in the active document. It can change the Master page on the active page only, a range of pages, or all pages that use a specified Page tag.

Syntax	Description
<b>.MasterPage</b>	Specifies the name of a Master page in the active document to apply.
<b>.FromPage</b>	Specifies the first page in a range to apply the specified Master page to. This parameter must be used with the <b>.ToPage</b> parameter. If this parameter is omitted, the specified Master page is applied to the active page.
<b>.ToPage</b>	Specifies the last page in a range to apply the specified Master page to. This parameter must be used with the <b>.FromPage</b> parameter. If this parameter is omitted, the specified Page tag is applied to the active page.
<b>.LeftPages</b>	If the <b>.FromPage</b> and <b>.ToPage</b> parameters are used, specifies whether to apply the Master Page to left pages in double-sideed documents. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is TRUE.
<b>.RightPages</b>	If the <b>.FromPage</b> and <b>.ToPage</b> parameters are used, specifies whether to apply the Master Page to right pages in double-sideed documents. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is TRUE.
<b>.OldMasterPage</b>	Specifies a Master Page in the active document. Any pages that use this Page tag have the <b>.MasterPage</b> parameter applied to them. If this parameter is used, the previous four parameters are ignored.

## Note

 This command corresponds to the Apply Master Page dialog box in Corel VENTURA. Click Page, Apply Master Page.

## Example

```
.FormatMasterPageApply "Intro"
```

The above example applies the Intro Master page to the active page.

```
.FormatMasterPageApply "Intro", 14, 66, TRUE, TRUE
```

The above example applies the Intro Master page to pages 14 through 66.

```
.FormatMasterPageApply "Intro", , , , "Setup"
```

The above example applies the Intro Master page to all pages using the Setup Page tag.


---

{button ,AL("VENTURA\_MasterPage;;;','0,"Defaultoverview",)} [Related Topics](#)



# .FormatMasterPageGet (VENTURA)

**.FormatMasterPageGet** .Portrait=*Boolean*, .PaperType=*long*, .Width=*long*, .Height=*long*, .MasterPage=*string*

This function returns properties of the active Page tag (Master Page).

Syntax	Description
.Portrait	Specifies a numeric variable that is assigned the page orientation: TRUE (-1) for portrait and FALSE (0) for landscape.
.PaperType	Specifies a numeric variable that is assigned a number corresponding to the paper type. Click  for a complete list of paper types.
.Width	Specifies a numeric variable that is assigned the paper width in tenths of a micron.
.Height	Specifies a numeric variable that is assigned the paper height in tenths of a micron.
.MasterPage	Specifies a string variable that is assigned the name of the active Page tag.

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

## Example

```
DIM P_L AS BOOLEAN
.FormatMasterPageGet P_L, PT&, PW&, PH&, MP$
```

The above examples assigns the properties of the active Page tag to the variables **P\_L**, **PT**, **PW**, **PH**, and **MP**


**.FormatPageLayout (VENTURA)**

**.FormatPageLayout** .LayoutType=*long*, .Portrait=*Boolean*, .Width=*long*, .Height=*long*, .ShowHeader=*Boolean*, .ShowFooter=*Boolean*



This command sets the orientation, size, and layout style of a page or page tag.

To change a page tag's attributes, this command must be enclosed with the **.FormatPageTagBegin** and **.FormatPageTagEnd** commands.

To change a selected page's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.LayoutType	Specifies the paper type. The following is a list of some common paper types: 0 Letter 1 Legal 2 Tabloid 10 A4 11 A5 34 Custom  Click  for a complete list of paper types.
.Portrait	Specifies the page orientation. Set to TRUE (-1) for portrait. Set to FALSE (0) for landscape. If omitted, the default is TRUE.
.Width	Required parameter if <b>.PaperType</b> is set to 34 (Custom). Specifies the paper width in tenths of a micron.
.Height	Required parameter if <b>.PaperType</b> is set to 34 (Custom). Specifies the paper height in tenths of a micron.
.ShowHeader	Specifies whether to show or hide page header. Set to TRUE (-1) to enable this option. If omitted, it is set to FALSE (0).
.ShowFooter	Specifies whether to show or hide page footer. Set to TRUE (-1) to enable this option. If omitted, it is set to FALSE (0).

**Note**

-  This command was introduced in Corel VENTURA 8 and replaces **FormatMasterPage**.
-  This command corresponds to the Page Properties and Page Tag Properties dialog box in Corel VENTURA. Click Page menu in Corel VENTURA for more information.

**Example**

```
.FormatObjectBegin 3
.FormatPageLayout 0, TRUE
.FormatObjectEnd
```

The above example sets the active page to a portrait orientation using a letter size paper type.


```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatPageTagBegin "Sample"
.FormatPageLayout 34, TRUE, 5*M_INCH, 7*M_INCH
.FormatPageTagEnd
```

The above example sets the Sample page tag to a portrait orientation using a paper size of 5 by 7 inches. The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

## .FormatPageLayoutGet (VENTURA)

**.FormatPageLayoutGet** .LayoutType=*long* , .Portrait=*Boolean* , .Width=*long* , .Height=*long* , .ShowHeader=*Boolean* , .ShowFooter=*Boolean*

This function returns the layout attributes for either the active page or for a specified page tag. To specify a page tag, enclose the function with the **.FormatPageTagBegin** and **.FormatPageTagEnd** commands.

Syntax	Description
.LayoutType	Specifies a numeric variable that is assigned the paper type. The following is a list of some common paper types: 0 Letter 1 Legal 2 Tabloid 10 A4 11 A5 34 Custom  Click  for a complete list of paper types.
.Portrait	Specifies a numeric variable that is assigned page orientation: TRUE (-1) for portrait and FALSE (0) for landscape.
.Width	Specifies a numeric variable that is assigned paper width in tenths of a micron.
.Height	Specifies a numeric variable that is assigned paper height in tenths of a micron.
.ShowHeader	Specifies a numeric variable that is assigned a value indicating whether the page header is visible or hidden: TRUE (-1) if visible; otherwise FALSE (0).
.ShowFooter	Specifies a numeric variable that is assigned a value indicating whether the page footer is visible or hidden: TRUE (-1) if visible; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This function was introduced in Corel VENTURA 8.

### Example

```
DIM orient AS BOOLEAN
.FormatPageLayoutGet LO&, orient
```

In the above example, the function assigns the active page's attributes to the variables **LO** and **orient**.

To use this example to return a specified page tag's attributes (the Sample page tag), use the following syntax:

```
.FormatPageTagBegin "Sample"
    DIM orient AS BOOLEAN
    .FormatPageLayoutGet LO&, orient
.FormatPageTagEnd
```

---

{button ,AL('VENTURA\_MasterPage;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.FormatPageTagBegin (VENTURA)**

**.FormatPageTagBegin** .TagName=*string*, .TagsInSelection=*Boolean*

Together, the **FormatPageTagBegin** and **FormatPageTagEnd** commands are required to change a page tag's attributes. The two commands enclose Corel VENTURA commands that can modify a page tag. The **.FormatPageTagBegin** statement is placed before the first page modifying command and the **FormatPageTagEnd** command is placed after the last page modifying command. This command was introduced in Corel VENTURA 8.

Syntax	Description
.TagName	Specifies the page tag to modify. The tag must already exist in the document. This parameter is required if the <b>TagsInSelection</b> parameter is omitted.  Multiple tags can be specified by separating them with semicolons. For example, "Tag1; Tag2; Tag3".
.TagsInSelection	Specifies whether to modify the tags in the current selection. Set to TRUE (-1) to enable this option; otherwise FALSE (0) which is the default if omitted. This parameter is mutually exclusive with the <b>TagName</b> parameter and takes precedence when set to TRUE. This parameter was introduced in Corel VENTURA 8.

### **Note**

The following commands can be used to modify a page or page tag.



#### **FormatPageLayout command**

Frame and rule formatting commands can also be used to modify a page or page tag including:



#### **.FormatFrameColumns**



#### **.FormatFrameTypography**



#### **.FormatFrameMarginsInside**



#### **.FormatFrameColumnSet**



#### **.FormatFrameColGutterOutline**



#### **.FormatRuleStyle**



#### **.FormatRuleSide**

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatPageTagBegin "Sample"
    .FormatPageLayout 34, TRUE, 5*M_INCH, 7*M_INCH
.FormatPageTagEnd
```

In the above example, the **Sample** page tag has its layout attributes modified.

---

{button ,AL('VENTURA\_MasterPage;;;','0',"Defaultoverview",)} [Related Topics](#)



## **.FormatPageTagEnd (VENTURA)**

### **.FormatPageTagEnd**

Together, the **.FormatPageTagBegin** and **.FormatPageTagEnd** commands are required to change a page tag's attributes. The two commands enclose VENTURA commands that can modify a page tag. The **.FormatPageTagBegin** statement is placed before the first page modifying command and the **.FormatPageTagEnd** command is placed after the last page modifying command. This command was introduced in Corel VENTURA 8.

#### **Note**

The following commands can be used to modify a page tag.



#### **.FormatPageLayout command**

Frame and rule formatting commands can also be used to modify a page or page tag including:



#### **.FormatFrameColumns**



#### **.FormatFrameTypography**



#### **.FormatFrameMarginsInside**



#### **.FormatFrameColumnSet**



#### **.FormatFrameColGutterOutline**



#### **.FormatRuleStyle**



#### **.FormatRuleSide**

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatPageTagBegin "Sample"
    .FormatPageLayout 34, TRUE, 5*M_INCH, 7*M_INCH
.FormatPageTagEnd
```

In the above example, the **Sample** page tag has its layout attributes modified.

---

{button ,AL('VENTURA\_MasterPage;;;','0',"Defaultoverview"),} [Related Topics](#)

## **.MasterPageAddNew (VENTURA)**


**.MasterPageAddNew .PageName=string, .CopyFrom=string**

This command is obsolete in Corel VENTURA 8. It has been replaced by [.PageTagAddNew](#).

This command adds a new Master Page to a document.

Syntax	Description
<b>.PageName</b>	Specifies the name of the new Master Page.
<b>.CopyFrom</b>	Specifies the name of the Master Page on which to base the formatting and properties of the new Master Page.

### **Note**

 This command corresponds to the Add command on the right mouse button menu in the Ventura Navigator. Click Tools, Navigator and choose Master Pages from the drop-down list box. Click the right mouse button, then click Add.

### **Example**

```
.MasterPageAddNew "Appendix", "Intro"
```

The above example adds the Master Page Appendix, which is based on the Master Page Intro.

---

**{button ,AL("VENTURA\_MasterPage;;;','0,"Defaultoverview",,)} [Related Topics](#)**

## **.MasterPageCopy (VENTURA)**


**.MasterPageCopy .PageName=string**

This command is obsolete in Corel VENTURA 8. It has been replaced by [PageTagCopy](#).

This command copies a Master Page tag from the active publication to the Clipboard. It can be used to copy Master Pages into VENTURA Libraries.

Syntax	Description
<b>.PageName</b>	Specifies the name of the Master Page to copy.

### **Note**

 This command corresponds to the Copy command for the Navigator. Click Tools, Navigator and choose Master Pages from the list box. Right-click, then click Copy.

### **Example**

```
.MasterPageCopy "BackPage"  
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste
```

The above example copies the Master Page BackPage, creates a VENTURA library named MYLIB.VLB, and pastes the Master Page into the new library.

---

{button ,AL('VENTURA\_MasterPage;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.MasterPageCount (VENTURA)**

**ReturnValue**& = .MasterPageCount ( )

This command is obsolete in Corel VENTURA 8. It has been replaced by [PageTagCount](#).

This function returns the number of Master Pages in a publication.

Syntax	Description
<b>ReturnValue</b>	Specifies a numeric variable that is assigned the number of Master Pages in the active document.

### **Note**



This command cannot be recorded.

### **Example**

```
Mcount& = .MasterPageCount
```

The above example assigns the number of Master Pages in the active document to the **Mcount** variable.

---

{button ,AL('VENTURA\_MasterPage;;;','0,"Defaultoverview",,)} [Related Topics](#)

## **.MasterPageDelete (VENTURA)**


**.MasterPageDelete .PageName=string, .ReformatPageName=string**


This command is obsolete in Corel VENTURA 8. It has been replaced by [PageTagDelete](#).

This command deletes a Master Page from a document.

Syntax	Description
<b>.PageName</b>	Specifies the name of the Master Page to delete.
<b>.ReformatPageName</b>	Specifies the Master Page to apply to pages using the Master Page specified in <b>.PageName</b> .

### **Note**

 This command corresponds to the Delete command in the right mouse button menu in the Ventura Navigator. Click Tools, Navigator and choose Master Pages from the drop-down list box. Click the right mouse button, then click Delete.

 You cannot delete the "Default Master" Master Page.

### **Example**

`.MasterPageDelete "Intro" , "TOC"`

The above example deletes the Master Page Intro and applies the TOC Master Page to pages using the Intro Master Page.

---

`{button ,AL('VENTURA_MasterPage;;;','0,"Defaultoverview",)} Related Topics`

## **.MasterPageGetAt (VENTURA)**

**ReturnString\$ = .MasterPageGetAt .PageIndex=*long***

This function is obsolete in Corel VENTURA 8. It has been replaced by [PageTagGetAt](#).

This function returns the name of a Master Page.

Syntax	Description
<b>ReturnString\$</b>	Specifies a string variable that is assigned the name of a Master Page.
<b>.PageIndex</b>	Specifies the index number of a Master Page. Index numbers are not associated with Master Page order in the VENTURA Navigator (which is in alphabetical order). Instead, index numbers are associated by the order in which the Master Pages are created. The default Master Page is 1, the first created Master Page is 2, and so on. The last created Master Page is equal to the return value from the <a href="#">.MasterPageCount</a> function. If a Master Page is deleted, the page index is recompiled.

### **Note**

 This command cannot be recorded.

### **Example**

```
Mcount& = .MasterPageCount
FOR i% = 1 TO Mcount&
    MPName$ = .MasterPageGetAt (i)
    MESSAGE MPName$
NEXT i%
```

The above example assigns the number of Master Pages in the active document to the **Mcount** variable. This variable is then used in a loop which displays the name of each Master Page in the active document in a message box.

---

**{button ,AL("VENTURA\_MasterPage;;;','0,"Defaultoverview"),} [Related Topics](#)**

## **.MasterPageRename (VENTURA)**


**.MasterPageRename .OldPageName=*string*, .NewPageName=*string***

This command is obsolete in Corel VENTURA 8. It has been replaced by [PageTagRename](#).

This command renames a specified Master Page.

Syntax	Description
<b>.OldPageName</b>	Specifies the Master Page to be renamed.
<b>.NewPageName</b>	Specifies the Master Page's new name.

### **Note**

 This command corresponds to the Rename command in the right mouse button menu in the Ventura Navigator. Click Tools, Navigator and choose Master Pages from the drop-down list box. Click the right mouse button, then click Rename.

### **Example**

`.MasterPageRename "Name 1" "Name 2".`

The above example renames the Master Page Name 1 to Name 2.

---

`{button ,AL('VENTURA_MasterPage;;;','0,"Defaultoverview",)} Related Topics`

## **.PageTagAddNew (VENTURA)**

**.PageTagAddNew** **.TagName=string**, **.CopyFrom=string**, **.FromSelectedPage=Boolean**

This command adds a new page tag to a publication.

Syntax	Description
<b>.TagName</b>	Specifies the name of the new page tag. This parameter is case sensitive.
<b>.CopyFrom</b>	Specifies the name of the page tag on which to base the formatting and properties of the new page tag.
<b>.FromSelectedPage</b>	Specifies whether to use the current page's page tag to base the new page tag on. Set TRUE (-1) to enable this option; otherwise FALSE (0) which is the default if omitted.

### **Note**



This command was introduced in Corel VENTURA 8 and replaces **.MasterPageAddNew**.



This command corresponds to the Tag Window. Click Tools, Tag Window.

### **Example**

```
.MasterPageAddNew "Appendix", "Intro"
```

The above example adds the Appendix page tag which is based on the Intro page tag.

---

**{button ,AL("VENTURA\_MasterPage;;;','0,"Defaultoverview",)} Related Topics**





# .PageTagApply (VENTURA)

**.PageTagApply** **.PageTag=string**, **.FromPage=long**, **.ToPage=long**, **.LeftPages=Boolean**, **.RightPages=Boolean**, **.OldPageTag=string**, **.CurrentChapter=Boolean**

This command changes the page tag on specified pages in the active publication. It can change the page tag on the active page only, a range of pages, or all pages that use a specified page tag.

Syntax	Description
<b>.PageTag</b>	Specifies the name of a Page tag in the active document to apply.
<b>.FromPage</b>	Specifies the first page in a range to apply the specified Page tag to. This parameter must be used with the <b>.ToPage</b> parameter. If this parameter is omitted, the specified Page tag is applied to the active page.
<b>.ToPage</b>	Specifies the last page in a range to apply the specified Page tag to. This parameter must be used with the <b>.FromPage</b> parameter. If this parameter is omitted, the specified Page tag is applied to the active page.
<b>.LeftPages</b>	If the <b>.FromPage</b> and <b>.ToPage</b> parameters are used, specifies whether to apply the Page tag to left pages in double-sideed documents. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is TRUE.
<b>.RightPages</b>	If the <b>.FromPage</b> and <b>.ToPage</b> parameters are used, specifies whether to apply the Page tag to right pages in double-sideed documents. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the default is TRUE.
<b>.OldPageTag</b>	Specifies a Page tag in the active document. Any pages that use this Page tag have the <b>.MasterPage</b> parameter applied to them. If this parameter is used, the previous four parameters are ignored.
<b>.CurrentChapter</b>	Specifies whether to apply the page tag to the pages in current chapter. Set TRUE (-1) to apply; otherwise set to FALSE (0) which is the default if omitted.

## Note

-  This command was introduced in Corel VENTURA 8 and replaces **FormatMasterPageApply**.
-  This command corresponds to the Apply Page Tag dialog box in Corel VENTURA. Click Page, Apply Page Tag.

## Example

`.PageTagApply "Intro"`

The above example applies the Intro Page tag to the active page.

`.PageTagApply "Intro", 14, 66, TRUE, TRUE`

The above example applies the Intro Page tag to pages 14 through 66.

`.PageTagApply "Intro", , , , "Setup"`

The above example applies the Intro Page tag to all pages using the Setup Page tag.

---

`{button ,AL('VENTURA_MasterPage;;;',0,"Defaultoverview"),}` [Related Topics](#)

## .PageTagCopy (VENTURA)

**.PageTagCopy .TagName=string**

This command copies a page tag from the active publication to the Clipboard. It can be used to copy page tags into a VENTURA Library or into another publication.

Syntax	Description
<b>.TagName</b>	Specifies the name of the page tag to copy. This parameter is case sensitive.

### Note



This command was introduced in Corel VENTURA 8 and replaces **MasterPageCopy**.



This command corresponds to the Tag Window. Click Tools, Tag Window.

### Example

```
.PageTagCopy "BackPage"  
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste
```

The above example copies the BackPage page tag, creates a VENTURA Library named MYLIB.VLB, and pastes the page tag into the new library.

---

**{button ,AL("VENTURA\_MasterPage;;;','0,"Defaultoverview",)} Related Topics**

## .PageTagCount (VENTURA)

**ReturnValue**& = .PageTagCount

This function returns the number of page tags in publication.

Syntax	Description
<b>ReturnValue</b>	Specifies a numeric variable that is assigned the number of page tags in the active publication.

### Note

 This function was introduced in Corel VENTURA 8 and replaces **MasterPageCount**.

### Example

```
Mcount& = .PageTagCount
```

The above example assigns the number of Page tags in the active document to the **Mcount** variable.

---

{button ,AL('VENTURA\_MasterPage;;;','0,"Defaultoverview",,)} [Related Topics](#)

## .PageTagDelete (VENTURA)

**.PageTagDelete .TagName=string, .PageTagName=string**

This command deletes a page tag from a publication.

Syntax	Description
<b>.TagName</b>	Specifies the name of the page tag to delete. This parameter is case sensitive.
<b>.PageTagName</b>	Specifies the page tag to apply to pages using the page tag specified in <b>.TagName</b> . This parameter is case sensitive.

### Note



This command was introduced in Corel VENTURA 8 and replaces MasterPageDelete.



This command corresponds to the Tag Window. Click Tools, Tag Window.



You cannot delete the Default Page page tag.

### Example

```
.PageTagDelete "Intro" , "TOC"
```

The above example deletes the Intro page tag and applies the TOC page tag to pages using the Intro page tag.

---

{button ,AL('VENTURA\_MasterPage;;;','0',"Defaultoverview",,)} Related Topics

## .PageTagGetAt (VENTURA)

**ReturnString\$ = .PageTagGetAt .TagIndex=long**

This function returns the name of a page tag.

Syntax	Description
<b>ReturnString\$</b>	Specifies a string variable that is assigned the name of a page tag.
<b>.TagIndex</b>	Specifies the index number of a page tag. Index numbers are not associated with page tag order in the Tag Window. Instead, index numbers are associated by the order in which the page tag are created. The default page tag is 1, the first created page tag is 2, and so on. The last created page tag is equal to the return value from the <a href="#">.PageTagCount</a> function. If a page tag is deleted, the page index is recompiled.

### Note



This function was introduced in Corel VENTURA 8 and replaces [MasterPageGetAt](#).

### Example

```
Mcount& = .PageTagCount
FOR i% = 1 TO Mcount&
    MPName$ = .PageTagGetAt (i)
    MESSAGE MPName$
NEXT i%
```

The above example assigns the number of page tags in the active document to the **Mcount** variable. This variable is then used in a loop which displays the name of each page tag in the active publication in a message box.

---

{button ,AL("VENTURA\_MasterPage;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.PageTagRename (VENTURA)**

**.PageTagRename .OldTagName=*string*, .NewTagName=*string***

This command renames a specified page tag.

<b>Syntax</b>	<b>Description</b>
<b>.OldTagName</b>	Specifies the page tag to be renamed. This parameter is case sensitive.
<b>.NewTagName</b>	Specifies the page tag's new name.

### **Note**



This command was introduced in Corel VENTURA 8 and replaces **MasterPageRename**.



This command corresponds to the Tag Window. Click Tools, Tag Window.

### **Example**

`.PageTagRename "Name 1" "Name 2".`

The above example renames the Name 1 page tag to Name 2.

---

`{button ,AL('VENTURA_MasterPage;;;','0,"Defaultoverview",)} Related Topics`

## **.DuplicateObject (VENTURA)**

**.DuplicateObject** .Direction=*long*, .Count=*long*

This command duplicates the currently selected object.

<b>Syntax</b>	<b>Description</b>
.Direction	Specifies the direction of duplication. 0      left (default if omitted) 1      right 2      up 3      down
.Count	Specifies the number of times to duplicate the object. The default is 1.

### **Example**

`.DuplicateObject 0, 3`

This example duplicates the currently selected object three times to the left.

---

{button ,AL('VENTURA\_Object;;;;','0,"Defaultoverview",')} [Related Topics](#)

## .FormatSetCurrentColor (VENTURA)

**.FormatSetCurrentColor** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This command sets the color for the active selection (text, table cells, frames, and graphics).

Syntax	Description
.ColorModel	Specifies the color model to apply to the selection: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
.Color1	Specifies the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.
.Color3	Specifies the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.
.Color4	Specifies the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the specified color model, omit it.

### Note



This command cannot be recorded.



The **.FillUniform** and the **.FormatSelectedTextColor** commands can also be used to apply colors to objects and text in VENTURA.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatSetCurrentColor
.FormatSelectedText 5, 0, 30, 240
```

The above example sets the selected text to a dark blue color using the RGB color model. The last parameter is omitted because RGB only uses three color components.

The above example draws a rectangle and an ellipse. The rectangle is then selected using the **.SelectObjectAt** command and is set to a dark blue color using the RGB color model. The last parameter is omitted because RGB only uses three color components.

The **.LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Object;;;;','0,"Defaultoverview",')} [Related Topics](#)




## .GetSelectionType (VENTURA)

ReturnValue& = .GetSelectionType ( )

This function returns the type of the current selection.

Syntax	Description
ReturnValue&	Specifies the numeric variable that is assigned the type of the current selection:
0	No selection in selection mode (arrow mouse pointer)
1	No selection in text mode (I-beam insertion point)
2	Selected text
3	Frame(s)
4	Graphic object(s)
5	Frame(s) and graphic(s) (multiple selection)

### Note

 This command cannot be recorded.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
GST& = .GetSelectionType ( )
```

The above example selects an object at a specified point and assigns the object's type to the **GST** variable.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Object;;;;','0,"Defaultoverview",)} Related Topics

## .MoveObject (VENTURA)

**.MoveObject .AmountX=*long*, .AmountY=*long***

This command moves the currently selected frame or graphic to another position on the active page.

Syntax	Description
<b>.AmountX</b>	Specifies the distance to horizontally move the left side of the selected object's bounding box in tenths of a micron. Positive values move it to the right; negative values move to it the left.
<b>.AmountY</b>	Specifies the distance to vertically move the top side of the selected object's bounding box in tenths of a micron. Positive values move it down; negative values move it up.

### Note



If more than one frame or graphic is selected, this command moves the first selected object.



The [.GraphicPosition](#) command can also be used to move a graphic.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.EditDuplicate
.MoveObject 1.5*M_INCH, 3.5*M_INCH
```

The above example draws an ellipse and duplicates it. The duplicate ellipse is then moved to the left by 1.5 inches and down by 3.5 inches.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Object;;;;','0,"Defaultoverview"),} [Related Topics](#)

## .NudgeObject (VENTURA)

**.NudgeObject** .Direction=*long*, .Count=*long*

This command nudges the currently selected object(s).

Syntax	Description
.Direction	Specifies the direction of nudge: 0 left 1 right (default if omitted) 2 up 3 down
.Count	Specifies the number of times to nudge. The default is 1 if omitted.

### Note

**I** The nudge distance is set in the Options dialog box. To edit this distance in Corel VENTURA click Tools, Options and in the Workspace category click Selection.

### Example

`.NudgeObject 0, 5`

This example nudges the currently select object five times to the left.

---

{button ,AL("VENTURA\_Object;;;;','0,"Defaultoverview",)} Related Topics

## .ResizeObject (VENTURA)

**.ResizeObject .ScaleX=*long*, .ScaleY=*long***

This command resizes the currently selected frame or graphic by scaling the selected object's width and height.

Syntax	Description
<b>.ScaleX</b>	Specifies the amount to scale the selected object's bounding box width. This amount is expressed in thousandths of a percent (100,000 equals 100 percent). Values greater than 100,000 increase the width; values less than 100,000 decrease the width.
<b>.ScaleY</b>	Specifies the amount to scale the selected object's bounding box height. This amount is expressed in thousandths of a percent (100,000 equals 100 percent). Values greater than 100,000 increase the height; values less than 100,000 decrease the height.

### Note



If more than one frame or graphic is selected, this command scales the first selected object.



The [.GraphicPosition](#) command can also be used to resize a graphic.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.EditDuplicate
.ResizeObject 50000, 75000
```

The above example draws an ellipse and duplicates it. The duplicate ellipse has its width scaled by 50% and its height by 75%. The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch. The following examples double and halve the size of a frame or graphic:

```
'double
.ResizeObject 200000, 200000
```

```
'halve
.ResizeObject 50000, 50000
```

---

**{button ,AL("VENTURA\_Object;;;;";0,"Defaultoverview",)} [Related Topics](#)**

# .RotateObject (VENTURA)

.RotateObject .Angle=*long*

This command rotates a selected frame or graphic around its center of rotation pin.

Syntax	Description
.Angle	Specifies the angle of rotation in tenths of degrees (45 degrees equals 450 tenths of a degree), relative to the center of the graphic. Positive numbers result in counter-clockwise rotation; negative numbers result in clockwise rotation.

Note

- If more than one frame or graphic is selected, this command rotates first selected object.
- The .GraphicRotate command can also be used to rotate a graphic.

Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.RotateObject 450
```

The above example draws a rectangle 3 by 4 inches. The rectangle is then rotated 45 degrees from a specified center of rotation position.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

{button ,AL("VENTURA\_Object;;;;','0,"Defaultoverview"),} Related Topics

## **.SelectNextObject (VENTURA)**

**.SelectNextObject .Previous=*Boolean*, .Extend=*Boolean***

This command selects the next or previous graphic object or frame on the current page.

Syntax	Description
<b>.Previous</b>	Set to TRUE (-1) to select the previous object. The default is FALSE (0).
<b>.Extend</b>	Set to TRUE (-1) to extend the selection. The default is FALSE (0).

### **Example**

```
.SelectNextObject TRUE, TRUE
```

This example selects the previous graphic object or frame on the current page and extends the selection.

---

{button ,AL('VENTURA\_Object;;;;','0',"Defaultoverview",,)} [Related Topics](#)

## .SelectObjectAt (VENTURA)

**.SelectObjectAt** *.PointX=long, .PointY=long, .Extend=Boolean*

This command toggles the selection of a graphic or frame at a specified point in the active page.

Syntax	Description
<b>.PointX</b>	Specifies the distance from the left side of the <u>base page frame</u> to an X-coordinate (horizontal position) within the object you want to select, in tenths of a micron.
<b>.PointY</b>	Specifies the distance from the top side of the <u>base page frame</u> to an Y-coordinate (vertical position) within the object you want to select, in tenths of a micron.
<b>.Extend</b>	If an object(s) is already selected, specifies whether to extend the selection. Set to TRUE (-1) to extend the selection. Set to FALSE (0) to de-select the currently selected object(s). If omitted, this parameter is set to FALSE.

### Note



If more than one object exists at the specified coordinates, the top-most object is selected.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4.5*M_INCH, 5*M_INCH, 1.5*M_INCH, 1*M_INCH
.SelectObjectAt 2.5*M_INCH, 3*M_INCH, FALSE
```

The above example draws a rectangle and an ellipse. The rectangle is then selected using the **.SelectObjectAt** command. The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawRectangle 2*M_INCH, 2*M_INCH, 3*M_INCH, 4*M_INCH
.DrawEllipse 4*M_INCH, 4*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.SelectObjectAt 2*M_INCH, 2*M_INCH, TRUE
.EditGroup
.RotateObject 450
.EditUngroup
```

The above example draws a rectangle and an ellipse. While the ellipse is already selected, the rectangle is also selected. Both objects are then grouped as a single object and rotated 45 degrees. The objects are then ungrouped.

---

{button ,AL('VENTURA\_Object;;;;','0,"Defaultoverview",)} Related Topics

## .EditObject (VENTURA)

**.EditObject .InPlace=Boolean**

This command activates the server of the currently selected OLE object to enable editing of the object.

Syntax	Description
<b>.InPlace</b>	<p>Specifies whether to edit the OLE object using the OLE object's in place server or edit the OLE object by starting the OLE object's server application. Set to TRUE (-1) to activate the in place server. Set to FALSE (0) to open the server application which is the default if omitted.</p> <p>When this parameter is set to TRUE, the <b>.EditObject</b> command corresponds to the Edit command in the Edit menu. Click Edit, <i>Object Name</i> Object, Edit.</p> <p>When this parameter is set to FALSE, the <b>.EditObject</b> command corresponds to the Open command in the Edit menu. Click Edit, <i>Object Name</i> Object, Open.</p>

### Note



This command cannot be recorded.



This command is available only when an OLE object is selected.

### Example

```
.InsertObject "C:\win95\LEAF.bmp", TRUE  
.EditObject TRUE
```

The above example inserts the LEAF bitmap into the active document as an OLE object. The bitmap is then opened for editing by its OLE server application within Corel VENTURA.

---

{button ,AL(^vent\_ole;;;',0,"Defaultoverview"),} Related Topics



## .EditObjectInfo (VENTURA)

**.EditObjectInfo** .DisplayAsIcon=*Boolean*, .AutomaticLink=*Boolean*, .BreakLink=*Boolean*

This command sets the properties of a selected OLE object.

Syntax	Description
.DisplayAsIcon	Specifies whether the selected OLE object appears as an icon or whether the OLE object is visible. Set to TRUE (-1) to display as an icon; otherwise, set to FALSE (0) which is the default if omitted. This corresponds to clicking Edit, <i>Object Name</i> Object, Convert, Display As Icon.
.AutomaticLink	If the selected OLE object is linked, specifies how the OLE object is updated. Set to TRUE (-1) for automatic updating; otherwise set to FALSE (0) for manual updating which is the default if omitted. This corresponds to clicking Edit, Links, and then clicking Automatic or Manual.
.BreakLink	If the selected OLE object is linked, set to TRUE (-1) to break its link. This corresponds to clicking Edit, Links, Break Link.

### Note

**I** This command cannot be recorded.

### Example

.EditObjectInfo TRUE, FALSE

The above example sets the properties for a selected OLE object so that it is displayed as an icon and it is manually updated.


---

{button ,AL('vent\_ole;;;;',0,"Defaultoverview",)} Related Topics



**.EditObjectInfoGet (VENTURA)**

**.EditObjectInfoGet** *.SourceName=string , .ObjectType=long , .DisplayAsIcon=Boolean , .AutomaticLink=Boolean*

This function returns the properties of a selected OLE object.

Syntax	Description
.SourceName	Specifies a string variable that is assigned the OLE object's server name. Click  for some examples of OLE object server names. An empty string is assigned if the selected object is not an OLE object.
.ObjectType	Specifies a numeric variable that is assigned a value corresponding to how the OLE object is embedded: 0      Non-OLE object 1      Linked 2      Embedded 3      Static (OLE object that has its link broken)
.DisplayAsIcon	Specifies a variable that is assigned a value indicating whether the OLE object is displayed as an icon: TRUE (-1) if displayed as an icon; otherwise, FALSE (0).
.AutomaticLink	Specifies a variable that is assigned a value indicating how the OLE object is linked: TRUE (-1) if automatically linked; otherwise, FALSE (0) if manually linked. If the OLE object is not linked (that is, the <b>.ObjectType</b> variable is not assigned 1), FALSE (0) is assigned.

**Note**

-  This command cannot be recorded.
-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

**Example**

```
DIM D_ICON AS BOOLEAN
DIM A_LINK AS BOOLEAN
.EditObjectInfoGet S_NAME$, OT&, D_ICON, D_ALINK
```

The above example assigns the properties of a selected OLE object to the variables **S\_NAME**, **OT**, **D\_ICON**, and **D\_ALINK**.

## **.EditPasteLink (VENTURA)**

### **.EditPasteLink**

This command pastes the contents of the Clipboard to the active document and establishes a link to the original document where the Clipboard contents were first created.

#### **Note**



You can also use the **.EditPaste** and **.EditPasteSpecial** commands to paste the Clipboard contents.

#### **Example**

' Perform some work in another application and copy it  
' to the Clipboard  
.EditPasteLink

The above example creates a link between the original application that created data and the active document.

---

{button ,AL('vent\_ole;;;','0,"Defaultoverview",,)} **Related Topics**

## .EditPasteSpecial (VENTURA)

**.EditPasteSpecial** .Format=*long*, .PasteLink=*Boolean*, .Pastelcon=*Boolean*

This command pastes or links the Clipboard contents into the active document using a specified format. The original cut or copied object remains on the Clipboard until you cut or copy another object or end the current Windows session.

If the pasted object is embedded, it is pasted into the document but retains its application identity. If the object is linked, it retains its identity and can be updated in the originating application with the results reflected in the linked object.

Syntax	Description
.Format	<p>Specifies the format of the Clipboard contents to paste:</p> <ul style="list-style-type: none"><li>-1 Automatic (default if omitted)</li><li>0 OLE object</li><li>1 Bitmap</li><li>2 DIB</li><li>3 Metafile picture</li><li>4 Text</li><li>5 RTF</li><li>6 BUDI</li><li>7 BUDI 16</li><li>8 VENTURA text</li><li>9 VENTURA graphic</li><li>10 VENTURA tag</li><li>11 VENTURA file</li><li>12 VENTURA table</li></ul> <p>If the <b>.Format</b> parameter is set to Automatic (-1) or omitted, Corel VENTURA attempts to paste the Clipboard contents using one of the following formats (in the following order).</p> <p>Ventura graphic BUDI BUDI 16 Text VENTURA table VENTURA text RTF VENTURA tag VENTURA file OLE object Metafile picture DIB Bitmap</p>
.PasteLink	<p>Specifies whether the pasted object is linked to the original document. Changes you make to a linked object are reflected in the original document, and changes in the original document also appear in the document where the object was pasted. Set to TRUE (-1) to link; otherwise set to FALSE (0). This parameter is ignored if the <b>.Format</b> parameter uses a format that doesn't support linking.</p>
.Pastelcon	<p>Specifies whether the pasted object appears in the document as an icon or as a visible object. If you display the object as an icon, you can open the object in your document by double-clicking its icon. Set to TRUE (-1) to paste as an icon; otherwise, set to FALSE (0). This parameter is ignored if the <b>.Format</b> parameter uses a format that doesn't support pasting as an icon.</p>



This command cannot be recorded.



This command corresponds to the Paste Special command in the Edit menu in Corel VENTURA. Click Edit, Paste Special.



If no parameters are specified, the Paste Special dialog box opens.



You can also use the **.EditPaste** and **.EditPasteLink** commands to paste the Clipboard contents.

### Example

' We know that the Clipboard contains a bitmap from the  
' Windows Paintbrush.  
EditPasteSpecial 1, FALSE, TRUE

The bitmap image format is the first format in the list and is used to paste an object into a document as an icon.

---

{button ,AL(^vent\_ole;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.EditUpdateNow (VENTURA)**

### **.EditUpdateNow**

This command updates the information in a selected linked OLE object from its original source.

#### **Note**



This command corresponds to the Links dialog box. Click Edit, Links, Update Now.

#### **Example**

.EditUpdateNow


---

{button ,AL('vent\_ole;;;','0,"Defaultoverview",,)} Related Topics

## .InsertObject (VENTURA)

**.InsertObject** **.SourceName=string**, **.FromFile=Boolean**, **.DisplayAsIcon=Boolean**, **.Link=Boolean**

This command inserts an OLE object into a document as a linked or embedded item.

Syntax	Description
<b>.SourceName</b>	Specifies the object to insert. This parameter can either specify an existing file name or a new OLE object. A new OLE object is specified using the OLE object's server name. Click  for some examples of OLE object server names.
<b>.FromFile</b>	Specifies whether <b>.SourceName</b> specifies a file or an OLE object. Set to TRUE (-1) to specify a file. Set to FALSE (0) to specify a new OLE object.
<b>.DisplayAsIcon</b>	Specifies whether to display the new OLE object as an icon in the VENTURA document. Set to TRUE (-1) to display as icon; otherwise, set to FALSE (0) which is the default if omitted.
<b>.Link</b>	Specifies whether to link or embed the OLE object. Set to TRUE (-1) to link; otherwise set to FALSE (0) which is the default if omitted.

### Note



This command cannot be recorded.



This command corresponds to the Insert Object dialog box in Corel VENTURA. To insert a file, click Insert, Object, Create From File. To insert an new OLE object, click Insert, Object, Create New.

### Example

**.InsertObject** "C:\win95\LEAF.bmp", TRUE

The above example inserts the LEAF bitmap into the active document as an OLE object.

**.InsertObject** "Paint.Picture", FALSE

The above example inserts a Windows bitmap into the active document as an OLE object.

---

**{button ,AL("vent\_ole;;;','0,"Defaultoverview"),}** [Related Topics](#)

## **.OleItemCopy (VENTURA)**

### **.OleItemCopy**

This command copies a selected OLE item to the Clipboard.

#### **Note**



This command was introduced in Corel VENTURA 8.

#### **Example**

`.OleItemCopy`

---

`{button ,AL('vent_ole;;;','0,"Defaultoverview",,)} Related Topics`



## .EditManageOverrides (VENTURA)

**.EditManageOverrides** .Alignment=*Boolean*, .Breaks=*Boolean*, .Hyphenation=*Boolean*, .Indents=*Boolean*, .FontAttributes=*Boolean*, .Effects=*Boolean*, .RulingLine=*Boolean*, .Spacing=*Boolean*, .Tabs=*Boolean*, .Typography=*Boolean*, .All=*Boolean*, .CopyToTag=*string*

This command removes the overrides in the current paragraph or copies paragraph attributes in the current paragraph to other paragraph tags.

Syntax	Description
.Alignment	Specifies whether to remove or transfer alignment attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.Breaks	Specifies whether to remove or transfer break attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.Hyphenation	Specifies whether to remove or transfer hyphenation attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.Indents	Specifies whether to remove or transfer indent attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.FontAttributes	Specifies whether to remove or transfer font attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.Effects	Specifies whether to remove or transfer drop cap or bullet attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.RulingLine	Specifies whether to remove or transfer ruling line attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.Spacing	Specifies whether to remove or transfer spacing attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.Tabs	Specifies whether to remove or transfer tab attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.Typography	Specifies whether to remove or transfer typography attributes. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.All	Specifies whether to remove or transfer all the attributes noted above. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If this option is enabled, all the above parameter settings are ignored.
.CopyToTag	Specifies the name of the existing tag to copy the specified attributes to. The specified attributes can include overridden and non-overridden attributes. This specified tag is also applied to the current paragraph.  If this parameter is omitted, the <b>.EditManageOverrides</b> command removes the specified overriding attributes from the current paragraph.

### Note



This command cannot be recorded.



This command corresponds to the Manage Overrides command in the Edit menu in Corel VENTURA. Click Edit, Manage Overrides.

### Example

```
.EditManageOverrides TRUE, , , , TRUE, , , , , MainHeading
```

The above example copies the alignment and effects attributes in the current paragraph to the MainHeading paragraph tag.

```
.EditManageOverrides TRUE, , , TRUE, , TRUE
```

The above example removes alignment, indent, and effects override attributes from the current paragraph.

---

{button ,AL("VENTURA\_FormatParaAlignment\_Menu;vent\_para;;;",0,"Defaultoverview",)} [Related Topics](#)

## **.FormatGetParaTag (VENTURA)**

**ReturnString\$ = .FormatGetParaTag ( )**

This function returns the paragraph tag name of the paragraph containing the insertion point.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of the active paragraph's tag.

### **Note**



This command cannot be recorded.



You can also use the **.FrameTagGetAt** command to find the name of a frame tag.

### **Example**

```
CurrentParaTag$ = .FormatGetParaTag ( )
```

In the above example, the paragraph tag name of the active paragraph is assigned to the **CurrentParaTag** string variable.

---

{button ,AL("VENTURA\_FormatParaAlignment\_Menu;vent\_para;;;','0,"Defaultoverview",)} Related Topics

## **.FormatParaAlignment (VENTURA)**

**.FormatParaAlignment** *.Horizontal=long*, *.Vertical=long*, *.FrameWideText=Boolean*, *.RightPageInFromLeft=long*, *.RightPageInFromRight=long*, *.LeftPageInFromLeft=long*, *.LeftPageInFromRight=long*, *.LinesToIndent=long*, *.LineIndentAmount=long*, *.AddPreviousLineWidth=Boolean*, *.DecimalInFromRight=long*, *.DecimalCharacter=long*, *.AbsoluteYPosition=long*

This command sets the paragraph indentation and alignment attributes for the active paragraph or changes the attributes of a paragraph tag.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Horizontal	Specifies the paragraph's horizontal position within its frame, column or page: 0 Left 1 Center 2 Right 3 Decimal 4 Justified (Last Line Left) 5 Justified (Last Line Justified) 6 Justified (Last Line Right) 7 Justified (Last Line Centered) These settings were modified in Corel VENTURA 8.
.Vertical	Specify the paragraph's vertical position within its frame, column or page: 0 Top 1 Middle 2 Bottom 3 Absolute Top is the usual alignment for text on the base page. Using Middle or Bottom on the base page forces succeeding paragraphs to the next page or column.
.FrameWideText	Specifies whether to have the paragraph span all columns in a multi-column layout. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the parameter is set to 0.
.RightPageInFromLeft	On the right-hand page, specifies the space between the left side of the paragraph and the left margin, in tenths of micron.
.RightPageInFromRight	On the right-hand page, specifies the space between the right side of the paragraph and the right margin, in tenths of micron.
.LeftPageInFromLeft	On the left-hand page, specifies the space between the left side of the paragraph and the left margin, in tenths of micron.
.LeftPageInFromRight	On the left-hand page, specifies the space between the right side of the paragraph and the right margin, in tenths of micron.
.LinesToIndent	Specifies the number of lines to indent.
.LinesIndentAmount	Specifies the amount of space between the left margin and the indented line(s), in tenths of micron. To make the first line hang out to the left the rest of the paragraph, specify a negative value.
.AddPreviousLineWidth	Specifies whether to indent the first line of the paragraph by an amount equal to the width of the last line of the previous paragraph. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.DecimalInFromRight	For decimal aligned paragraphs, specifies the distance from the right margin to the alignment point, in tenths of micron.
.DecimalCharacter	Specifies the decimal alignment character. The character is specified using ANSI character values. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters. Some of the most common characters used, and their ANSI values are: 46 . (period) 44 , (comma) 45 - (hyphen) 95 _ (underscore)

.AbsoluteYPosition

If **.Vertical** is set to Absolute, specifies the distance from the top of the page to the paragraph in tenths of a micron.

#### Note



This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

#### Example

```
.FormatObjectBegin 1  
.FormatParaAlignment 0, 1, , , , , 2  
.FormatObjectEnd
```

The above example sets the paragraph attributes to Left horizontal alignment, Top vertical alignment, and 2 lines to indent.

---

{button ,AL('VENTURA\_FormatParaAlignment\_Menu;vent\_para;;;','0,"Defaultoverview",')} Related Topics

## .FormatParaAlignmentGet (VENTURA)

**.FormatParaAlignmentGet** .Horizontal=*long*, .Vertical=*long*, .FrameWideText=*Boolean*, .RightPageInFromLeft=*long*, .RightPageInFromRight=*long*, .LeftPageInFromLeft=*long*, .LeftPageInFromRight=*long*, .LinesToIndent=*long*, .LineIndentAmount=*long*, .AddPreviousLineWidth=*Boolean*, .DecimalInFromRight=*long*, .DecimalCharacter=*long*, .AbsoluteYPosition=*long*

This function returns the alignment attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description																
.Horizontal	<p>Specifies the numeric variable that is assigned the paragraph's horizontal position attribute:</p> <table><tr><td>0</td><td>Left</td></tr><tr><td>1</td><td>Center</td></tr><tr><td>2</td><td>Right</td></tr><tr><td>3</td><td>Decimal</td></tr><tr><td>4</td><td>Justified (Last Line Left)</td></tr><tr><td>5</td><td>Justified (Last Line Justified)</td></tr><tr><td>6</td><td>Justified (Last Line Right)</td></tr><tr><td>7</td><td>Justified (Last Line Centered)</td></tr></table> <p>These settings were modified in Corel VENTURA 8.</p>	0	Left	1	Center	2	Right	3	Decimal	4	Justified (Last Line Left)	5	Justified (Last Line Justified)	6	Justified (Last Line Right)	7	Justified (Last Line Centered)
0	Left																
1	Center																
2	Right																
3	Decimal																
4	Justified (Last Line Left)																
5	Justified (Last Line Justified)																
6	Justified (Last Line Right)																
7	Justified (Last Line Centered)																
.Vertical	<p>Specifies the numeric variable that is assigned the paragraph's vertical position attribute:</p> <table><tr><td>0</td><td>Top</td></tr><tr><td>1</td><td>Middle</td></tr><tr><td>2</td><td>Bottom</td></tr><tr><td>3</td><td>Absolute</td></tr></table>	0	Top	1	Middle	2	Bottom	3	Absolute								
0	Top																
1	Middle																
2	Bottom																
3	Absolute																
.FrameWideText	<p>Specifies the numeric variable that is assigned the paragraph spanning columns attribute: TRUE (-1) if spanning is enabled; otherwise FALSE (0).</p>																
.RightPageInFromLeft	<p>Specifies the numeric variable that is assigned the amount of space between the left side of the paragraph and the left margin, on the right-hand page, in tenths of micron.</p>																
.RightPageInFromRight	<p>Specifies the numeric variable that is assigned the amount of space between the right side of the paragraph and the right margin, on the right-hand page, in tenths of micron.</p>																
.LeftPageInFromLeft	<p>Specifies the numeric variable that is assigned the amount of space between the left side of the paragraph and the left margin, on the left-hand page, in tenths of micron.</p>																
.LeftPageInFromRight	<p>Specifies the numeric variable that is assigned the amount of space between the right side of the paragraph and the right margin, on the left-hand page, in tenths of micron.</p>																
.LinesToIndent	<p>Specifies the numeric variable that is assigned the number of lines to indent.</p>																
.LinesIndentAmount	<p>Specifies the numeric variable that is assigned the amount of space between the left margin and the indented line(s), in tenths of micron.</p>																
.AddPreviousLineWidth	<p>Specifies the numeric variable that is assigned the "Add width of preceding line attribute": TRUE (-1) if option enabled; otherwise FALSE (0).</p>																
.DecimalInFromRight	<p>Specifies the numeric variable that is assigned the distance from the right margin to the alignment point, in tenths of micron (for decimal aligned paragraphs).</p>																
.DecimalCharacter	<p>Specifies the numeric variable that is assigned the decimal alignment character. The alignment character is assigned as an ANSI value. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters.</p> <p>Some of the most common characters used, and their ANSI values are:</p> <table><tr><td>46</td><td>.</td><td>(period)</td></tr><tr><td>44</td><td>,</td><td>(comma)</td></tr><tr><td>45</td><td>-</td><td>(hyphen)</td></tr><tr><td>95</td><td>_</td><td>(underscore)</td></tr></table>	46	.	(period)	44	,	(comma)	45	-	(hyphen)	95	_	(underscore)				
46	.	(period)															
44	,	(comma)															
45	-	(hyphen)															
95	_	(underscore)															
.AbsoluteYPosition	<p>Specifies the numeric variable that is assigned the distance from the top of the page to the paragraph in tenths of a micron. This parameter is only valid if <b>.Vertical</b> is set to Absolute.</p>																

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatParaAlignmentGet Horz&, Vert&, , , , , , , , , DecChar&
```

In the above example, the function assigns the horizontal, vertical, and decimal character settings of the active paragraph tag to the variables **Horz**, **Vert**, and **DecChar**, respectively.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
    .FormatParaAlignmentGet Horz&, Vert&, , , , , , , , DecChar&  
.FormatParaTagEnd
```

---

{button ,AL('VENTURA\_FormatParaAlignmentGet\_Menu;vent\_para;;;',0,"Defaultoverview",,)} [Related Topics](#)

## **.FormatParaBreaks (VENTURA)**

**.FormatParaBreaks** .PageBreak=*long*, .ColumnBreak=*long*, .LineBreak=*long*, .InLineWithPrev=*Boolean*, .AllowBreakWithin=*Boolean*, .KeepWithNext=*Boolean*, .KeepWithPrev=*Boolean*


This command controls the paragraph's position relative to other paragraphs and whether two consecutive paragraphs should be kept together on the same page or in the same column. After a break, the next paragraph can begin on a new line, at the top of a new column, at the top of a new page, or at the top of the next left or right page. The break can occur before, after, or both before and after the paragraph.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.PageBreak	Specifies where the page break will occur in relationship to the paragraph: 0 None 1 Before 2 After 3 Before & After 4 Before/Until Left 5 Before/Until Right
.ColumnBreak	Specify where the column break will occur in relationship to the paragraph: 0 None 1 Before 2 After 3 Before & After
.LineBreak	Specify whether space is added between the paragraph and the one before or after it: 0 None 1 Before 2 After 3 Before & After
.InLineWithPrev	Determines the vertical position of the paragraph when no line break exists between it and the one preceding it. Set to TRUE (-1) to place the beginning of the paragraph on the same line as the first line of the previous paragraph. Set to FALSE (0) to place the paragraph next to the last line.
.AllowBreakWithin	Controls whether breaks are allowed within the paragraph when there is not enough room for the entire paragraph on a page. Set to TRUE (-1) to allow page breaks; otherwise, set to FALSE (0).
.KeepWithNext	Specifies whether to keep the selected paragraph and the one following it together on the same page (or column). Set to TRUE (-1) to enable this option; otherwise, set to FALSE.
.KeepWithPrev	Specifies whether to keep the selected paragraph and the one preceding it together on the same page (or column). Set to TRUE (-1) to enable this option; otherwise, set to FALSE.

### **Note**

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### **Example**

```
.FormatObjectBegin 1
.FormatParaBreaks 1, , , , , -1
.FormatObjectEnd
```

The above example set the paragraph's attributes for a page break before the paragraph and enables breaks within the paragraph.

---

**{button ,AL("VENTURA\_FormatParaBreaks\_Menu;vent\_para;;;",0,"Defaultoverview"),} [Related Topics](#)**

**.FormatParaBreaksGet (VENTURA)**

**.FormatParaBreaksGet** *.PageBreak=long, .ColumnBreak=long, .LineBreak=long, .InLineWithPrev=Boolean, .KeepWithNext=Boolean, .KeepWithPrev=Boolean*

This function returns the break attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.PageBreak	Specifies the numeric variable that is assigned the paragraph's page break attribute: 0 None 1 Before 2 After 3 Before & After 4 Before/Until Left 5 Before/Until Right
.ColumnBreak	Specifies the numeric variable that is assigned the paragraph's column break attribute: 0 None 1 Before 2 After 3 Before & After
.LineBreak	Specifies the numeric variable that is assigned the paragraph's line break attribute: 0 None 1 Before 2 After 3 Before & After
.InLineWithPrev	Specifies the numeric variable that is assigned the paragraph vertical position attribute when no line break exists between it and the one preceding it: TRUE (-1) if the beginning of the paragraph is placed on the same line as the first line of the previous paragraph; FALSE (0) if the paragraph is placed next to the last line.
.AllowBreakWithin	Specifies the numeric variable that is assigned the break within paragraph attribute: TRUE (-1) if this option is enabled; otherwise FALSE (0).
.KeepWithNext	Specifies the numeric variable that is assigned the keep with next paragraph attribute: TRUE (-1) if this option is enabled; otherwise FALSE (0).
.KeepWithPrev	Specifies the numeric variable that is assigned the keep with previous paragraph attribute: TRUE (-1) if this option is enabled; otherwise FALSE (0).

**Note**

- I** The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
- I** This command cannot be recorded.

**Example**

```
.FormatParaBreaksGet , CBreak&
```

In the above example, the function assigns the active paragraph tag's column break attribute to the variable **CBreak**.  
To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
  .FormatParaBreaksGet , CBreak&  
.FormatParaTagEnd
```



## .FormatParaBullet (VENTURA)

**.FormatParaBullet** .Indent=*long*, .BulletCharacter=*long*, .ShiftUp=*long*, .IndentFirstLineOnly=*Boolean*

This command sets the paragraph bullet attributes for the active paragraph or changes the attributes of a paragraph tag. To use a bullet with a paragraph, see the [.FormatParaEffects](#) command.

To change a paragraph tag's attributes, this command must be enclosed with the [.FormatParaTagBegin](#) and [.FormatParaTagEnd](#) commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the [.FormatObjectBegin](#) and [.FormatObjectEnd](#) commands.

Syntax	Description								
.Indent	Specifies the indentation distance between the bullet character and the paragraph text, in tenths of a micron.								
.BulletCharacter	<p>Specifies the bullet character. The character is specified using ANSI character values. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters.</p> <p>Some of the most common characters used for bullets, and their ANSI values are:</p> <table><tr><td>183</td><td>bullet</td></tr><tr><td>61</td><td>equal sign</td></tr><tr><td>45</td><td>hyphen</td></tr><tr><td>95</td><td>underscore</td></tr></table>	183	bullet	61	equal sign	45	hyphen	95	underscore
183	bullet								
61	equal sign								
45	hyphen								
95	underscore								
.ShiftUp	Specifies the vertical shift of the bullet character, in tenths of a micron.								
.IndentFirstLineOnly	Specifies whether to indent only the first line. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).								

### Note

**I** This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 1
.FormatParaBullet .5*M_INCH, 183, .1*M_INCH, -1
.FormatObjectEnd
```

The above example sets the paragraph to use ANSI character number 183 for the bullet symbol. The bullet is set to be indented by 0.5 inches and shifted up by 0.1 inches. The indent is for the first line in the paragraph only.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatParaBullet\_Menu;vent\_para;;;',0,"Defaultoverview"),} [Related Topics](#)

## .FormatParaBulletGet (VENTURA)

**.FormatParaBulletGet** .Indent=*long*, .BulletCharacter=*long*, .ShiftUp=*long*, .IndentFirstLineOnly=*Boolean*

This function returns the bullet setting attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.Indent	Specifies the numeric variable that is assigned the indentation distance between the bullet character and the paragraph text, in tenths of a micron.
.BulletCharacter	Specifies the numeric variable that is assigned the bullet character. The bullet character is assigned as an ANSI value. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters.  Some of the most common characters used for bullets and their ANSI values are: 183   bullet 61     equal sign 45     hyphen 95     underscore
.ShiftUp	Specifies the numeric variable that is assigned the vertical shift of the bullet character, in tenths of a micron..
.IndentFirstLineOnly	Specifies the numeric variable that is assigned the bullet's first line indentation attribute: TRUE (-1) if only the first line is indented; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatParaBulletGet , AnsiNum&
```

In the above example, the function returns the active paragraph tag's bullet symbol number to the **AnsiNum** variable.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
    .FormatParaBulletGet , AnsiNum&  
.FormatParaTagEnd
```

---

{button ,AL('VENTURA\_FormatParaBulletGet\_Menu;vent\_para;;;',0,"Defaultoverview",)} [Related Topics](#)


## .FormatParaColor (VENTURA)

**.FormatParaColor** .Background=*Boolean*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .Overprint=*Boolean*, .ColumnWide=*Boolean*


This command sets the paragraph color attributes for the active paragraph, or changes the attributes of a paragraph tag.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Background	Specifies whether to change attributes for the background or font colors. Set to TRUE (-1) to set background colors. Set to FALSE (0) to set font colors.
.ColorModel	<p>Specifies the color model to use for the paragraph:</p> <ol style="list-style-type: none"><li>1 <u>PANTONE Spot Colors</u></li><li>2 <u>CMYK</u></li><li>3 <u>CMYK255</u></li><li>4 <u>CMY</u></li><li>5 <u>RGB</u> (default if omitted)</li><li>6 <u>HSB</u></li><li>7 <u>HLS</u></li><li>8 <u>Black and White</u></li><li>9 <u>Grayscale</u></li><li>11 <u>YIQ</u></li><li>12 <u>L*a*b*</u></li><li>14 <u>PANTONE Hexachrome</u></li></ol> <p>If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.</p>
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Overprint	<p>When the <b>.Background</b> parameter is set to TRUE, specifies whether to have the text's background print over objects beneath it. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0).</p> <p>When the <b>.Background</b> parameter is set to FALSE, specifies whether to have the text print over the background color. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0). This option is relevant only when printing color separations.</p>
.ColumnWide	When the <b>.Background</b> parameter is set to TRUE, specifies whether to extend the background to the edge of the column. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0).

### Note

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### Example

```
.FormatObjectBegin 1
.FormatParaColor -1, 5, 0, 30, 240
.FormatObjectEnd
```

The above example sets the paragraph background to a dark blue color using the RGB color model. The last parameter is omitted because RGB only uses 3 color components.

---

{button ,AL('VENTURA\_FormatParaColor\_Menu;vent\_para;;;','0,"Defaultoverview",)} Related Topics

## .FormatParaColorGet (VENTURA)

**.FormatParaColorGet** .Background=*Boolean*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .Overprint=*Boolean*, .ColumnWide=*Boolean*

This function returns the color attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.Background	Specifies whether to return the background or font color paragraph attributes. Set to TRUE (-1) to return background colors. Set to FALSE (0) to return font colors. If omitted, this setting is set to FALSE.
.ColorModel	Specifies the numeric variable that is assigned a return value corresponding to selected text's color mode: <div><div>1</div><div>PANTONE Spot Colors</div></div> <div><div>2</div><div>CMYK</div></div> <div><div>3</div><div>CMYK255</div></div> <div><div>4</div><div>CMY</div></div> <div><div>5</div><div>RGB (default if omitted)</div></div> <div><div>6</div><div>HSB</div></div> <div><div>7</div><div>HLS</div></div> <div><div>8</div><div>Black and White</div></div> <div><div>9</div><div>Grayscale</div></div> <div><div>11</div><div>YIQ</div></div> <div><div>12</div><div>L*a*b*</div></div> <div><div>14</div><div>PANTONE Hexachrome</div></div>
.Color1	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color3	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color4	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Overprint	Specifies the numeric variable that is assigned a value indicating whether the overprint attribute is enabled: TRUE (-1) if enabled; otherwise FALSE (0).
.ColumnWide	Specifies the numeric variable that is assigned a value indicating whether the extend background to the edge of the column option is enabled: TRUE (-1) if enabled; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatParaColorGet -1, cm&, c1&, c2&, c3&, c4&
```

The function the assigns the active paragraph tag's background color settings to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
  .FormatParaColorGet -1, cm&, c1&, c2&, c3&, c4&  
.FormatParaTagEnd
```

## .FormatParaDefaults (VENTURA)

**.FormatParaDefaults** *.AutoAdjust=Boolean*, *.ColumnsToSpan=long*, *.NextTag=string*, *.OverscoreThickness=long*, *.OverscoreShift=long*, *.StrikeThickness=long*, *.StrikeShift=long*, *.Underline1Thickness=long*, *.Underline1Shift=long*, *.Underline2Thickness=long*, *.Underline2Shift=long*, *.SuperscriptPointSize=single*, *.SuperscriptShift=long*, *.SubscriptPointSize=single*, *.SubscriptShift=long*, *.Tracking=long*, *.TrackAmount=long*

This command sets the default text attributes for the active paragraph or changes the attributes of a paragraph tag. Text attributes include items such as underline thickness, subscript character size, and character tracking.


To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.AutoAdjust	Specifies whether to automatically update all Default settings when the paragraph font size is changed. Changes will be proportionate to paragraph font size. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.ColumnsToSpan	Specifies the number of columns underlines, overscores and strike-thrus to extend.
.NextTag	Specifies the name of the paragraph tag VENTURA applies to the next paragraph when you press ENTER. Use this parameter when you want the next paragraph after the current one to always use a different tag. For example, after a heading you can have VENTURA automatically format the next paragraph with the Body Text tag.
.OverscoreThickness	Specifies the thickness for the overscore line, in tenths of a micron. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.OverscoreShift	Specifies the distance between the overscore and the baseline of the paragraph, in tenths of a micron. A positive value moves the overscore away from the baseline; a negative value moves it closer. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.StrikeThickness	Specifies the thickness for the strike-thru line, in tenths of a micron. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.StrikeShift	Specifies the distance between the strike-thru line and the baseline of the paragraph, in tenths of a micron. A positive value moves the strike-through line away from the baseline; a negative value moves it closer. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.Underline1Thickness	Specifies the thickness for underline 1, in tenths of a micron. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.Underline1Shift	Specifies the distance between underline 1 and the baseline of the paragraph, in tenths of a micron. A positive value moves the underline away from the baseline; a negative value moves it closer. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.Underline2Thickness	Specifies the thickness for underline 2, in tenths of a micron. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.Underline2Shift	Specifies the distance between underline 2 and the baseline of the paragraph, in tenths of a micron. A positive value moves the underline away from the baseline; a negative value moves it closer. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.SuperscriptPointSize	Specifies the a font size for the superscript character, in tenths of a micron. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.SuperscriptShift	Specifies the distance between the superscript character the baseline of the paragraph, in tenths of a micron. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.SubscriptPointSize	Specifies the a font size for the subscript character, in tenths of a micron. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.
.SubscriptShift	Specifies the distance between the subscript character the baseline of the paragraph, in tenths of a micron. The <b>.AutoAdjust</b> parameter must be set to FALSE (0), or else this parameter is ignored.

.Tracking	Specifies the tracking setting which is the spacing between characters in a paragraph:
0	Very loose
1	Looser
2	Normal
3	Tighter
4	Very tight
5	Custom
.TrackAmount	If the <b>.Tracking</b> parameter is set to Custom, specifies the tracking setting in tenths of an em. For example, 10.5 ems is equal to 105 tenths of an em. An em is a unit of measure roughly equal to the width of the capital M at the current font size.

#### Note

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

#### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 1
.FormatParaDefaults 0, , , .02*M_INCH, , , , .03*M_INCH
.FormatObjectEnd
```

The above example sets the overscore thickness to 0.02 inches and the underline 1 thickness to .03 inches. The auto adjust setting is set to FALSE (0).

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_FormatParaDefaults\_Menu;vent\_para;;;','0,"Defaultoverview",)} Related Topics

## **.FormatParaDefaultsGet (VENTURA)**

**.FormatParaDefaults** *.AutoAdjust=Boolean*, *.ColumnsToSpan=long*, *.NextTag=string*, *.OverscoreThickness=long*, *.OverscoreShift=long*, *.StrikeThickness=long*, *.StrikeShift=long*, *.Underline1Thickness=long*, *.Underline1Shift=long*, *.Underline2Thickness=long*, *.Underline2Shift=long*, *.SuperscriptPointSize=single*, *.SuperscriptShift=long*, *.SubscriptPointSize=single*, *.SubscriptShift=long*, *.Tracking=long*, *.TrackAmount=long*

This function returns the default attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.AutoAdjust	Specifies the numeric variable that is assigned the Auto Adjust attribute: TRUE (-1) if this option is enabled; otherwise FALSE (0).
.ColumnsToSpan	Specifies the numeric variable that is assigned the number of columns that underlines, overscores and strike-thrus to extend.
.NextTag	Specifies the string variable that is assigned the paragraph tag VENTURA applies when you press ENTER.
.OverscoreThickness	Specifies the numeric variable that is assigned the thickness for the overscore line, in tenths of a micron.
.OverscoreShift	Specifies the numeric variable that is assigned the distance between the overscore and the baseline of the paragraph, in tenths of a micron.
.StrikeThickness	Specifies the numeric variable that is assigned the thickness for the strike-thru line, in tenths of a micron.
.StrikeShift	Specifies the numeric variable that is assigned the distance between the strike-thru line and the baseline of the paragraph, in tenths of a micron.
.Underline1Thickness	Specifies the numeric variable that is assigned the thickness for underline 1, in tenths of a micron.
.Underline1Shift	Specifies the numeric variable that is assigned the distance between underline 1 and the baseline of the paragraph, in tenths of a micron.
.Underline2Thickness	Specifies the numeric variable that is assigned the thickness for underline 2, in tenths of a micron.
.Underline2Shift	Specifies the numeric variable that is assigned the distance between underline 2 and the baseline of the paragraph, in tenths of a micron.
.SuperscriptPointSize	Specifies the numeric variable that is assigned the a font size for the superscript character, in tenths of a micron.
.SuperscriptShift	Specifies the numeric variable that is assigned the distance between the superscript character the baseline of the paragraph, in tenths of a micron.
.SubscriptPointSize	Specifies the numeric variable that is assigned the a font size for the subscript character, in tenths of a micron.
.SubscriptShift	Specifies the numeric variable that is assigned the distance between the subscript character the baseline of the paragraph, in tenths of a micron.
.Tracking	Specifies the numeric variable that is assigned the tracking setting. Tracking is the spacing between characters in a paragraph: 0      Very loose 1      Looser 2      Normal 3      Tighter 4      Very tight 5      Custom
.TrackAmount	If the <b>.Tracking</b> parameter is set to Custom, specifies the numeric variable that is assigned the tracking setting in tenths of an em. For example, 10.5 ems is equal to 105 tenths of an em. An em is a unit of measure roughly equal to the width of the capital M at the current font size.

### **Note**

- 1 The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
- 1 This command cannot be recorded.

### **Example**



```
DIM AA AS BOOLEAN
.FormatParaDefaultsGet AA, , xtag$, othick&
```

In the above example, the function assigns the active paragraph tag's automatic adjustment, next tag, and overscore thickness attributes to the variables **AA**, **xtag**, and **othick**, respectively.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"
    .FormatParaDefaultsGet AA, , xtag$, othick&
.FormatParaTagEnd
```

---

{button ,AL('VENTURA\_FormatParaDefaultsGet\_Menu;vent\_para;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.FormatParaDropCap (VENTURA)**

**.FormatParaDropCap** .CharacterCount=*long*, .ShiftUp=*long*, .CustomLineCount=*long*, .CountType=*long*


This command sets the paragraph drop cap attributes for the active paragraph or changes the attributes of a paragraph tag. To use a drop cap with a paragraph, see the **.FormatParaEffects** command.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.CharacterCount	Specifies the number of characters or words to drop or raise in the paragraph.
.ShiftUp	Specifies the distance in tenths of micron from the baseline to shift the drop cap character. Positive numbers shift the character up; negative numbers shift it down.
.CustomLineCount	Specifies the number of lines to drop the character(s).
.LengthType	Specifies the setting for the <b>.CharacterCount</b> parameter: 0 characters 1 words

### **Note**

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FormatObjectBegin 1
.FormatParaDropCap 3, .1*M_INCH, 4
.FormatObjectEnd
```

The above example sets the number of drop cap characters to 3 in the paragraph. The drop caps are baseline shifted by 0.1 inches and dropped four lines.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_FormatParaDropCap\_Menu;vent\_para;;;',0,"Defaultoverview",)} **Related Topics**



# .FormatParaDropCapGet (VENTURA)

**.FormatParaDropCapGet** .CharacterCount=*long*, .ShiftUp=*long*, .CustomLineCount=*long*, .CountType=*long*

This function returns the drop cap setting attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.CharacterCount	Specifies the numeric variable that is assigned the number of characters or words to drop or raise in the paragraph.
.ShiftUp	Specifies the numeric variable that is assigned the distance in tenths of micron from the baseline to shift the drop cap character. Positive numbers shift the character up; negative numbers shift it down.
.CustomLineCount	Specifies the numeric variable that is assigned the number of lines to drop the character(s).
.LengthType	Specifies the numeric variable that is assigned a value indicating whether the <b>.CharacterCount</b> parameter is set to characters or words: 1 if words; 0 if characters.

### Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

### Example

```
.FormatParaDropCapGet cc&, , clc&
```

In the above example the function assigns the active paragraph tag's number of characters to drop or raise setting, and the number of lines to drop the characters setting, to the variables **cc**, and **clc**, respectively.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
    .FormatParaDropCapGet cc&, , clc&  
.FormatParaTagEnd
```


## .FormatParaEffectColor (VENTURA)

**.FormatParaEffectColor** .Background=*Boolean*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .Overprint=*Boolean*


This command sets the effect color attributes for the active paragraph or changes the attributes of a paragraph tag. The effect font is used with bullets and drop caps.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Background	Specifies whether to change attributes for the effect background or the effect font colors. Set to TRUE (-1) to set background colors. Set to FALSE (0) to set font colors.
.ColorModel	<p>Specifies the color model to use for the paragraph:</p> <ol style="list-style-type: none"><li>1 <u>PANTONE Spot Colors</u></li><li>2 <u>CMYK</u></li><li>3 <u>CMYK255</u></li><li>4 <u>CMY</u></li><li>5 <u>RGB</u> (default if omitted)</li><li>6 <u>HSB</u></li><li>7 <u>HLS</u></li><li>8 <u>Black and White</u></li><li>9 <u>Grayscale</u></li><li>11 <u>YIQ</u></li><li>12 <u>L*a*b*</u></li><li>14 <u>PANTONE Hexachrome</u></li></ol> <p>If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.</p>
.Color1	Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.
.Color2	Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color3	Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Color4	Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.
.Overprint	<p>When the <b>.Background</b> parameter is set to TRUE, specifies whether to have the text's background print over objects beneath it. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).</p> <p>When the <b>.Background</b> parameter is set to FALSE, specifies whether to have the text print over the background color. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).</p> <p>This option is relevant only when printing color separations.</p>

### Note

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### Example

```
.FormatObjectBegin 1
.FormatParaColor 0, 5, 0, 30, 240
.FormatObjectEnd
```

The above example sets the paragraph effect font color to a dark blue color using the RGB color model. The last parameter is omitted because RGB only uses 3 color components.

---

{button ,AL(^VENTURA\_FormatParaEffectColor\_Menu;vent\_para;;;','0,"Defaultoverview",)} [Related Topics](#)



## .FormatParaEffectColorGet (VENTURA)

**.FormatParaEffectColorGet** .Background=*Boolean*, .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*, .Overprint=*Boolean*

This function returns the effect color attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.Background	Specifies whether to return the effect background or the effect font color paragraph attributes. Set to TRUE (-1) to return background colors. Set to FALSE (0) to return font colors. If omitted, it is set to FALSE.
.ColorModel	Specifies the numeric variable that is assigned a return value corresponding to selected text's color mode: <div><div>1</div><div>PANTONE Spot Colors</div></div> <div><div>2</div><div>CMYK</div></div> <div><div>3</div><div>CMYK255</div></div> <div><div>4</div><div>CMY</div></div> <div><div>5</div><div>RGB (default if omitted)</div></div> <div><div>6</div><div>HSB</div></div> <div><div>7</div><div>HLS</div></div> <div><div>8</div><div>Black and White</div></div> <div><div>9</div><div>Grayscale</div></div> <div><div>11</div><div>YIQ</div></div> <div><div>12</div><div>L*a*b*</div></div> <div><div>14</div><div>PANTONE Hexachrome</div></div>
.Color1	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color3	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color4	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Overprint	Specifies the numeric variable that is assigned a value indicating whether the overprint attribute is enabled: TRUE (-1) if enabled; otherwise FALSE (0).

### Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

### Example

```
.FormatEffectColorGet 0, cm&, c1&, c2&, c3&, c4&
```

This example assigns the active paragraph tag's effect font color settings to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
    .FormatEffectColorGet 0, cm&, c1&, c2&, c3&, c4&  
.FormatParaTagEnd
```

---

{button ,AL("VENTURA\_FormatParaEffectColorGet\_Menu;vent\_para;;;",0,"Defaultoverview",)} Related Topics

## .FormatParaEffectFont (VENTURA)

**.FormatParaEffectFont** .FontName=string, .PointSize=single, .Weight=long, .Italic=Boolean, .Underline=long, .StrikeThru=Boolean, .Overscore=Boolean

This command sets the effect font attributes for the active paragraph or changes the attributes of a paragraph tag. The effect font is used with bullets and drop caps.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description																		
.FontName	Specifies the name of the font. For a list of installed fonts, see the font box on the text or property toolbars.																		
.PointSize	Specifies the font size in points.																		
.Weight	<p>Specifies the font's weight setting (number of inked pixels per 1000 pixels). The weight setting ranges from 0 to 1000. Common values and their corresponding names include:</p> <table><tr><td>100</td><td>Thin</td></tr><tr><td>200</td><td>Extra Light, Ultra Light</td></tr><tr><td>300</td><td>Light</td></tr><tr><td>400</td><td>Normal, Regular</td></tr><tr><td>500</td><td>Medium</td></tr><tr><td>600</td><td>Semi Bold, Demi Bold</td></tr><tr><td>700</td><td>Bold</td></tr><tr><td>800</td><td>Extra Bold, Ultra Bold</td></tr><tr><td>900</td><td>Black, Heavy</td></tr></table> <p>This parameter is ignored if the font you're applying this setting to doesn't have the specified weight installed on your system. For example, Courier is normally available with only Normal (400) and Bold (700) weight settings. If you set the weight for a Courier font to a value other than 400 or 700, this parameter is ignored and the default setting is used.</p> <p>Some fonts may use different names than those shown above for the same weight settings. See a font dialog box for more information about the fonts you have installed on your system.</p>	100	Thin	200	Extra Light, Ultra Light	300	Light	400	Normal, Regular	500	Medium	600	Semi Bold, Demi Bold	700	Bold	800	Extra Bold, Ultra Bold	900	Black, Heavy
100	Thin																		
200	Extra Light, Ultra Light																		
300	Light																		
400	Normal, Regular																		
500	Medium																		
600	Semi Bold, Demi Bold																		
700	Bold																		
800	Extra Bold, Ultra Bold																		
900	Black, Heavy																		
.Italic	Specifies whether to apply italic formatting. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). You must have an italic version of the specified font installed on your system or this parameter is ignored.																		
.Underline	<p>Specifies the font's underline attributes:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Single</td></tr><tr><td>2</td><td>Double</td></tr></table>	0	None	1	Single	2	Double												
0	None																		
1	Single																		
2	Double																		
.Strikethru	Specifies the font's strike-thru setting. Set to TRUE (-1) to enable strike-thru formatting. Set to FALSE (0) to disable strike-thru formatting.																		
.Overscore	Specifies the font's overscore setting. Set to TRUE (-1) to enable overscore formatting. Set to FALSE (0) to disable italics formatting.																		

### Note

**I** This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### Example

```
.FormatObjectBegin 1
.FormatParaEffectFont "Symbol", , 2
.FormatObjectEnd
```

The above example changes the paragraph effect font to Symbol using the Bold style.

```
.FormatParaEffectFont , 10.5
```

The above example changes the paragraph effect font to 10.5 points.

---

{button ,AL('VENTURA\_FormatParaEffectFont\_Menu;vent\_para;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FormatParaEffectFontGet (VENTURA)

**.FormatParaEffectFontGet** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .Underline=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*

This function returns the effect font attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.FontName	Specifies a string variable that is assigned the name of the font.
.PointSize	Specifies a numeric variable that is assigned the font size in points.
.Weight	Specifies a numeric variable that is assigned the font's weight setting (number of inked pixels per 1000 pixels). The weight setting ranges from 0 to 1000. Common values and their corresponding names include: 100 Thin 200 Extra Light, Ultra Light 300 Light 400 Normal, Regular 500 Medium 600 Semi Bold, Demi Bold 700 Bold 800 Extra Bold, Ultra Bold 900 Black, Heavy
.Italic	Specifies a numeric variable that is assigned the font's italic setting: TRUE (-1) if enabled; otherwise FALSE (0).
.Underline	Specifies a numeric variable that is assigned the font's underline attributes: 0 None 1 Single 2 Double
.Strikethru	Specifies a numeric variable that is assigned the font's strike-thru setting: TRUE (-1) if strike-thru formatting is enabled; otherwise FALSE (0).
.Overscore	Specifies a numeric variable that is assigned the font's overscore setting: TRUE (-1) if overscore formatting is enabled; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatParaEffectFontGet Fname$, , Fstyle&
```

The above example assigned the active paragraph tag's effect font name to the **Fname** variable and the font style setting to the **FStyle** variable.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
    .FormatParaEffectFontGet Fname$, , Fstyle&  
.FormatParaTagEnd
```

---

{button ,AL('VENTURA\_FormatParaEffectFontGet\_Menu;vent\_para;;;',0,"Defaultoverview"),} [Related Topics](#)



## **.FormatParaEffects (VENTURA)**

**.FormatParaEffects** .EffectType=*long*, .TextAfter=*string*, .TextBefore=*string*


This command sets the effect and the text before and after attributes for the active paragraph or changes the attributes of a paragraph tag.


To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.EffectType	Specifies the type of paragraph effect: 0     None 1     Bullet 2     Drop Cap
.TextAfter	Specifies the identifier of a previously defined Text Before/After text string. The text string is inserted after the paragraph that is formatted with the current tag. To create a text string, use the <b>.FormatPubTextAdd</b> command.
.TextBefore	Specifies the identifier of a previously defined Text Before/After text string. The text string is inserted before the paragraph formatted with the current tag. To create a text string, use the <b>.FormatPubTextAdd</b> command.

### **Note**

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

 To use this command to switch from bullets to drop caps or from drop caps to bullets, precede this command with another **FormatParaEffects** command with **EffectType** set to 0.

### **Example**

```
.FormatObjectBegin 1
.FormatParaEffects 1, "Trailer", "Leader"
.FormatObjectEnd
```

In the example above, the paragraph attribute for effect is set to Bullet. Two strings are also specified for the text after and before the paragraph.

---

{button ,AL('Ventura\_FormatParaEffects\_Menu;vent\_para;;;',0,"Defaultoverview"),} [Related Topics](#)



**.FormatParaEffectsGet (VENTURA)**

**.FormatParaEffectsGet** .EffectType=*long*, .TextAfter=*string*, .TextBefore=*string*

This function returns the effect and the text before and after attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the [.FormatParaTagBegin](#) and [.FormatParaTagEnd](#) commands.

Syntax	Description
.EffectType	Specifies the numeric variable that is assigned the paragraph's effect attribute: 0     None 1     Bullet 2     Drop Cap
.TextAfter	Specifies a string variable that is assigned the identifier of a previously defined Text Before/After text string (if exists) to insert after the paragraph formatted with the current tag. To create a text string, the <a href="#">.FormatPubTextAdd</a> command.
.TextBefore	Specifies a string variable that is assigned the identifier of a previously defined Text Before/After text string (if exists) to insert before the paragraph formatted with the current tag. To create a text string, the <a href="#">.FormatPubTextAdd</a> command.

**Note**

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

**Example**

```
.FormatParaEffectsGet fx&, TA$
```

In the above example, the active paragraph tag's effects setting is assigned to the **fx** variable and the text after setting is assigned to the **TA** variable.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
  .FormatParaEffectsGet fx&, TA$  
.FormatParaTagEnd
```

## **.FormatParaFont (VENTURA)**

**.FormatParaFont** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .Underline=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*, .Uppercase=*Boolean*

This command sets the font attributes for the active paragraph or changes the attributes of a paragraph tag.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description																		
.FontName	Specifies the name of the font. For a list of installed fonts, see the font box on the text or property toolbars.																		
.PointSize	Specifies the font size in points.																		
.Weight	<p>Specifies the font's weight setting (number of inked pixels per 1000 pixels). The weight setting ranges from 0 to 1000. Common values and their corresponding names include:</p> <table><tr><td>100</td><td>Thin</td></tr><tr><td>200</td><td>Extra Light, Ultra Light</td></tr><tr><td>300</td><td>Light</td></tr><tr><td>400</td><td>Normal, Regular</td></tr><tr><td>500</td><td>Medium</td></tr><tr><td>600</td><td>Semi Bold, Demi Bold</td></tr><tr><td>700</td><td>Bold</td></tr><tr><td>800</td><td>Extra Bold, Ultra Bold</td></tr><tr><td>900</td><td>Black, Heavy</td></tr></table> <p>This parameter is ignored if the font you're applying this setting to doesn't have the specified weight installed on your system. For example, Courier is normally available with only Normal (400) and Bold (700) weight settings. If you set the weight for a Courier font to a value other than 400 or 700, this parameter is ignored and the default setting is used.</p> <p>Some fonts may use different names than those shown above for the same weight settings. See a font dialog box for more information about the fonts you have installed on your system.</p>	100	Thin	200	Extra Light, Ultra Light	300	Light	400	Normal, Regular	500	Medium	600	Semi Bold, Demi Bold	700	Bold	800	Extra Bold, Ultra Bold	900	Black, Heavy
100	Thin																		
200	Extra Light, Ultra Light																		
300	Light																		
400	Normal, Regular																		
500	Medium																		
600	Semi Bold, Demi Bold																		
700	Bold																		
800	Extra Bold, Ultra Bold																		
900	Black, Heavy																		
.Italic	Specifies whether to apply italic formatting. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). You must have an italic version of the specified font installed on your system, or else this parameter is ignored.																		
.Underline	<p>Specifies the font's underline attributes:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Single</td></tr><tr><td>2</td><td>Double</td></tr><tr><td>3</td><td>Word underline</td></tr></table>	0	None	1	Single	2	Double	3	Word underline										
0	None																		
1	Single																		
2	Double																		
3	Word underline																		
.StrikeThru	Specifies the font's strike-thru setting. Set to TRUE (-1) to enable strike-thru formatting. Set to FALSE (0) to disable strike-thru formatting.																		
.Overscore	Specifies the font's overscore setting. Set to TRUE (-1) to enable overscore formatting. Set to FALSE (0) to disable italics formatting.																		
.Uppercase	Specifies whether to use uppercase characters with the font. Set to TRUE (-1) to enable uppercase characters; otherwise, set to FALSE (0).																		

### **Note**

**1** This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### **Example**

```
.FormatObjectBegin 1
.FormatParaFont "AvantGarde", ,1
.FormatObjectEnd
```

The above example changes the paragraph font to AvantGarde using the Normal/Italic style.

```
.FormatParaFont , 10.5, , 2, , , -1
```

The above example changes the paragraph font to 10.5 points, using double underline and uppercase.

---

{button ,AL(^VENTURA\_FormatParaFont\_Menu;vent\_para;;;,0,"Defaultoverview",)} Related Topics

## .FormatParaFontGet (VENTURA)

**.FormatParaFontGet** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .Underline=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*, .Uppercase=*Boolean*

This function returns the font attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.FontName	Specifies a string variable that is assigned the name of the font.
.PointSize	Specifies a numeric variable that is assigned the font size in points.
.Weight	Specifies a numeric variable that is assigned the font's weight setting (number of inked pixels per 1000 pixels). The weight setting ranges from 0 to 1000. Common values and their corresponding names include: 100 Thin 200 Extra Light, Ultra Light 300 Light 400 Normal, Regular 500 Medium 600 Semi Bold, Demi Bold 700 Bold 800 Extra Bold, Ultra Bold 900 Black, Heavy
.Italic	Specifies a numeric variable that is assigned the font's italic setting: TRUE (-1) if enabled; otherwise FALSE (0).
.Underline	Specifies a numeric variable that is assigned the font's underline attributes: 0 None 1 Single 2 Double 3 Word underline
.Strikethru	Specifies a numeric variable that is assigned the font's strike-thru setting: TRUE (-1) if strike-thru formatting is enabled; otherwise FALSE (0).
.Overscore	Specifies a numeric variable that is assigned the font's overscore setting: TRUE (-1) if overscore formatting is enabled; otherwise FALSE (0).
.Uppercase	Specifies a numeric variable that is assigned the whether to use uppercase characters with the font: TRUE (-1) if uppercase characters is enabled; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM UnSet AS BOOLEAN
.FormatParaFontGet Fname$, , Fw&, , UnSet
```

The above example assigns the active paragraph tag's font name, weight setting, and underline setting to the **Fname**, **Fw**, and **UnSet** variables, respectively.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"
.FormatParaFontGet Fname$, , Fw&, , UnSet
.FormatParaTagEnd
```

---

{button ,AL('VENTURA\_FormatParaFontGet\_Menu;vent\_para;;;',0,"Defaultoverview",)} Related Topics

## .FormatParaHyphenation (VENTURA)

**.FormatParaHyphenation** .Automatic=*Boolean*, .Dictionary=*string*, .SuccessiveHyphens=*long*, .MinWordSize=*long*, .MinCharactersBefore=*long*, .MinCharactersAfter=*long*, .AtPageEnd=*Boolean*, .AtLastWord=*Boolean*, .ForCapitalized=*Boolean*


This command sets the hyphenation attributes for the active paragraph or changes the attributes of a paragraph tag.


To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.


To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.


Syntax	Description
.Automatic	Specifies whether to automatically hyphenate text in the paragraph using the dictionary specified in the <b>.Dictionary</b> parameter. Set to TRUE (-1) to enable this option, otherwise, set to FALSE (0).
.Dictionary	Specifies the language dictionary VENTURA uses when hyphenating the paragraph.
.SuccessiveHyphens	Specifies the maximum number of consecutive lines that can end with a hyphen. Set this parameter to 255 if you do not want to set a maximum. The <b>.AutoAdjust</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.MinWordSize	Specifies the minimum length of hyphenated words in the paragraph. For example, if you do not want words of less than five characters hyphenated, set to four. The <b>.AutoAdjust</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.MinCharactersBefore	Specifies the minimum number of characters that can precede a hyphen. The <b>.AutoAdjust</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.MinCharactersAfter	Specifies the minimum number of characters that can follow a hyphen. The <b>.AutoAdjust</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.AtPageEnd	Specifies whether to prevent hyphenation of the last word in a column or page. Set to TRUE (-1) to prevent hyphenation of the last word in a column or page; otherwise, set to FALSE (0). The <b>.AutoAdjust</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.AtLastWord	Specifies whether to prevent hyphenation of the last word in a paragraph. Set to TRUE (-1) to prevent hyphenation of the last word in a paragraph; otherwise, set to FALSE (0). The <b>.AutoAdjust</b> parameter must be set to TRUE (-1), or else this parameter is ignored.
.ForCapitalized	Specifies whether to prevent hyphenation of uppercase words such as a product name. Set to TRUE (-1) to prevent hyphenation of uppercase words; otherwise, set to FALSE (0). The <b>.AutoAdjust</b> parameter must be set to TRUE (-1), or else this parameter is ignored.

### Note

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

 You can override the automatic hyphenation at any time by inserting discretionary hyphens. However, automatic hyphenation must still be enabled for a paragraph for discretionary hyphens to work.

 The English language version of VENTURA installs two hyphenation dictionaries: American English and British English. For information on how to install another language dictionaries, see Corel VENTURA's main online help.

 If you always hyphenate a word differently than VENTURA, or don't want to hyphenate a particular word, you can customize the built-in hyphenation dictionary.

### Example

```
.FormatObjectBegin 1
.FormatParaHyphenation -1, "Italian", , 4
.FormatObjectEnd
```

The above example enables automatic hyphenation using the Italian dictionary and sets the minimum word to hyphenate to 4 characters.

---

{button ,AL("VENTURA\_FormatParaHyphenation\_Menu;vent\_para;;;",0,"Defaultoverview"),} [Related Topics](#)

## .FormatParaHyphenationGet (VENTURA)

**.FormatParaHyphenationGet** .Automatic=*Boolean*, .Dictionary=*string*, .SuccessiveHyphens=*long*, .MinWordSize=*long*, .MinCharactersBefore=*long*, .MinCharactersAfter=*long*, .AtPageEnd=*Boolean*, .AtLastWord=*Boolean*, .ForCapitalized=*Boolean*

This function returns the hyphenation attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.Automatic	Specifies the numeric variable that is assigned the automatic hyphenation attribute: TRUE (-1) if this option is enabled; otherwise FALSE (0).
.Dictionary	Specifies the numeric variable that is assigned the language dictionary VENTURA uses when hyphenating the paragraph.
.SuccessiveHyphens	Specifies the numeric variable that is assigned the maximum number of consecutive lines that can end with a hyphen.
.MinWordSize	Specifies the numeric variable that is assigned the minimum length of hyphenated words in the paragraph.
.MinCharactersBefore	Specifies the numeric variable that is assigned the minimum number of characters that can precede a hyphen.
.MinCharactersAfter	Specifies the numeric variable that is assigned the minimum number of characters that can follow a hyphen.
.AtPageEnd	Specifies the numeric variable that is assigned the a value corresponding to whether hyphenation of the last word in a column or page is prevented: TRUE (-1) if hyphenation is prevented; otherwise FALSE (0).
.AtLastWord	Specifies the numeric variable that is assigned the a value corresponding to whether hyphenation of the last word in paragraph is prevented: TRUE (-1) if hyphenation is prevented; otherwise FALSE (0).
.ForCapitalized	Specifies the numeric variable that is assigned the a value corresponding to whether hyphenation of capitalized words is prevented: TRUE (-1) if hyphenation is prevented; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM AH AS BOOLEAN
.FormatParaHyphenationGet AH, Dic$, , MinWord&
```

In the above example the function assigns the active paragraph tag's automatic hyphenation, dictionary, and minimum word size to the variables **AH**, **Dic**, and **MinWord**, respectively.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"
.FormatParaHyphenationGet AH, Dic$, , MinWord&
.FormatParaTagEnd
```

---

{button ,AL("VENTURA\_FormatParaHyphenationGet\_Menu;vent\_para;;;",0,"Defaultoverview",)} [Related Topics](#)

## **.FormatParaSpacing (VENTURA)**

**.FormatParaSpacing** *.SpaceAbove=long, .SpaceBelow=long, .SpaceInterLine=long, .SpaceInterPara=long, .AutoAdjustLine=long, .GrowInterline=Boolean, .AddAboveSpace=Boolean, .AddAboveSpaceOnly=Boolean, .RotationAngle=long, .RotationHeight=long*

This command sets the paragraph spacing and paragraph's angle of rotation attributes for the active paragraph or changes the attributes of a paragraph tag.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.SpaceAbove	Specifies the amount of space above the paragraph, in tenths of a micron. When the space above a paragraph and the space below the preceding one differ, VENTURA sets the spacing to the larger of the two.
.SpaceBelow	Specifies the amount of space below the paragraph, in tenths of a micron. When the space above a paragraph and the space below the preceding one differ, VENTURA sets the spacing to the larger of the two.
.SpaceInterLine	Specifies the amount of space between lines of text in the paragraph, in tenths of micron.
.SpaceInterPara	Specifies the amount of space between identical paragraphs, in tenths of a micron. This spacing only takes effect between paragraphs with identical Inter-Paragraph spacing. Inter Paragraph space is useful when you want the space between body paragraphs to be greater than the space between the headings and the body text.
.AutoAdjustLine	Specifies how to adjust spacing when the paragraph font size changes: 1     Percentage 2     Add 4     None
.GrowInterline	Specifies whether to increase inter-line spacing to accommodate characters larger than the normal paragraph font. Set to TRUE (-1) to adjust spacing; otherwise, set to FALSE (0).
.AddAboveSpace	Specifies whether to add spacing above the paragraph, even if the paragraph is at the top of a column or frame. Set to TRUE (-1) to add spacing; otherwise, set to FALSE (0). If omitted, the parameter is set to 0.  This setting must be enabled to increase space above header and footer text and text at the top of a caption or footnote frame.
.AddAboveSpaceOnly	Specifies whether to add spacing above the paragraph only when it appears at the top of a column or frame. Set to TRUE (-1) to add spacing; otherwise, set to FALSE (0). If omitted, the parameter is set to 0.  You might use this setting to insert additional space above headings at top of the pages.
.RotationAngle	Specifies paragraph's rotation angle in degrees.
.RotationHeight	Specifies the maximum amount of space the paragraph occupies, in tenths of a micron. If the line length of a paragraph rotated 90 or 270 degrees exceeds the limit, VENTURA wraps the text to the next line. For paragraphs rotated 180 degrees, this setting controls the vertical position.

### **Note**

**I** This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### **Example**

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatObjectBegin 1
.FormatParaSpacing 6*M_POINT, , 1*M_POINT,
.FormatObjectEnd
```

The above example sets the space above the paragraph to 6 points and the interline spacing in the paragraph to 1 point.

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in an point.

---

{button ,AL('VENTURA\_FormatParaSpacing\_Menu;vent\_para;;;',0,"Defaultoverview"),} [Related Topics](#)



## **.FormatParaSpacingGet (VENTURA)**

**.FormatParaSpacingGet** *.SpaceAbove=long, .SpaceBelow=long, .SpaceInterLine=long, .SpaceInterPara=long, .AutoAdjustLine=long, .GrowInterline=Boolean, .AddAboveSpace=Boolean, .AddAboveSpaceOnly=Boolean, .RotationAngle=long, .RotationHeight=long*

This function returns the spacing attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.SpaceAbove	Specifies the numeric variable that is assigned the amount of space above the paragraph, in tenths of a micron.
.SpaceBelow	Specifies the numeric variable that is assigned the amount of space below the paragraph, in tenths of a micron.
.SpaceInterLine	Specifies the amount of space between lines of text in the paragraph, in tenths of micron.
.SpaceInterPara	Specifies the numeric variable that is assigned the amount of space between identical paragraphs, in tenths of a micron.
.AutoAdjustLine	Specifies the numeric variable that is assigned the how to adjust spacing when the paragraph font size changes: 1     Percentage 2     Add 4     None
.GrowInterline	Specifies the numeric variable that is assigned a value corresponding to whether the paragraph increases inter-line spacing to accommodate characters larger than the normal paragraph font: TRUE (-1) if this option is enabled; otherwise FALSE (0).
.AddAboveSpace	Specifies the numeric variable that is assigned a value corresponding to whether the paragraph adds spacing above, even if the paragraph is at the top of a column or frame: TRUE (-1) if this option is enabled; otherwise FALSE (0).
.AddAboveSpaceOnly	Specifies the numeric variable that is assigned a value corresponding to whether the paragraph adds spacing above only when it appears at the top of a column or frame: TRUE (-1) if this option is enabled; otherwise FALSE (0).
.RotationAngle	Specifies the numeric variable that is assigned paragraph's rotation angle in tenths of a degree.
.RotationHeight	Specifies the numeric variable that is assigned the maximum amount of space the paragraph occupies, in tenths of a micron.

### **Note**



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### **Example**

```
.FormatParaSpacingGet SA&, SB&, , SI&
```

In the above example the function assigns the active paragraph tag's space above, space below, and inter-paragraph spacing to **SA**, **SB**, and **SI**, respectively.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
  .FormatParaSpacingGet SA&, SB&, , SI&  
.FormatParaTagEnd
```

---

{button ,AL('VENTURA\_FormatParaSpacingGet\_Menu;vent\_para;;;',0,"Defaultoverview"),} Related Topics

## .FormatParaTab (VENTURA)

**.FormatParaTab .TrailLeader=Boolean**, .TrailLeaderCharacter=*long*, .TrailLeaderSpacing=*long*


This command sets the paragraph trailing leader attributes for the active paragraph or changes the attributes of a paragraph tag. Trailing leaders fill the space between the end of the paragraph and the right margin with leader characters.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description																								
<b>.TrailLeader</b>	Specifies whether to enable trailing leaders. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0), which is the default if omitted.																								
.TrailLeaderCharacter	<p>Specifies the trailing leader character. The character is specified using ANSI character values. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters.</p> <p>Some of the most common characters used and their ANSI values are:</p> <table><tr><td>35</td><td># (pound sign)</td></tr><tr><td>42</td><td>* (asterisk)</td></tr><tr><td>44</td><td>, (comma)</td></tr><tr><td>45</td><td>- (hyphen)</td></tr><tr><td>46</td><td>. (period)</td></tr><tr><td>47</td><td>/ (forward slash)</td></tr><tr><td>60</td><td>&lt; (less than)</td></tr><tr><td>62</td><td>&gt; (greater than)</td></tr><tr><td>95</td><td>_ (underscore)</td></tr><tr><td>126</td><td>~ (tilde)</td></tr><tr><td>183</td><td>· (bullet)</td></tr><tr><td>187</td><td>» (double bracket)</td></tr></table> <p>This parameter is used only if the <b>.TrailLeader</b> parameter is enabled.</p>	35	# (pound sign)	42	* (asterisk)	44	, (comma)	45	- (hyphen)	46	. (period)	47	/ (forward slash)	60	< (less than)	62	> (greater than)	95	_ (underscore)	126	~ (tilde)	183	· (bullet)	187	» (double bracket)
35	# (pound sign)																								
42	* (asterisk)																								
44	, (comma)																								
45	- (hyphen)																								
46	. (period)																								
47	/ (forward slash)																								
60	< (less than)																								
62	> (greater than)																								
95	_ (underscore)																								
126	~ (tilde)																								
183	· (bullet)																								
187	» (double bracket)																								
.TrailLeaderSpacing	Specifies the number of spaces between leader characters. The maximum value is 9. This parameter is used only if the <b>.TrailLeader</b> parameter is enabled.																								

### Note

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### Example

```
.FormatObjectBegin 1
.FormatParaTab TRUE, 35, 4
.FormatObjectEnd
```

The above example enables the trailing leader attribute, using the pound sign character (#), and a spacing of 4.

---

{button ,AL('VENTURA\_FormatParaTab\_Menu;vent\_para;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.FormatParaTabClear (VENTURA)**

**.FormatParaTabClear** .Position=*long*, .All=*Boolean*

This command clears tab stops in the active paragraph or changes the tag attributes of a paragraph tag.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Position	Specifies the tab setting to clear. Tab settings are ordered based on their position value in inches. The smallest value in inches is the first ordered tab. The first tab setting is 1, the second tab setting is 2, and so on.
.All	Specifies whether to clear all tab stops. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). If omitted, the parameter is set to 0.

### **Note**



To set a tab, use the **.FormatParaTabSet** command.



This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### **Example**

```
.FormatObjectBegin 1
.FormatParaTabClear , -1
.FormatObjectEnd
```

The above example clears all the tabs in the paragraph.

```
.FormatParaTabClear 2
```

The above example clears the second tab.

---

{button ,AL('VENTURA\_FormatParaTabClear\_Menu;vent\_para;;;',0,"Defaultoverview"),} [Related Topics](#)


# .FormatParaTabCount (VENTURA)

ReturnValue& =.FormatParaTabCount ( )

This function returns the number of tab settings for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the .FormatParaTagBegin and .FormatParaTagEnd commands.

Syntax	Description
ReturnValue&	The numeric variable that is assigned the number of tab settings in the active paragraph.

Note

 This command cannot be recorded.

Example

```
TabsCount& = .FormatParaTabCount ( )
```

The above example assigns the number of tab settings in the active paragraph to the **TabsCount** variable.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
    TabsCount& = .FormatParaTabCount ( )  
.FormatParaTagEnd
```

## .FormatParaTabGet (VENTURA)

**.FormatParaTabGet** **.TrailLeader=Boolean**, **.TrailLeaderCharacter=long**, **.TrailLeaderSpacing=long**

This function returns the trailing leader attributes for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
<b>.TrailLeader</b>	Specifies the variable that is assigned the trailing leader status: TRUE (-1) if enabled; otherwise, FALSE (0).
<b>.TrailLeaderCharacter</b>	Specifies the numeric variable that is assigned the trailing leader character as an ANSI character value. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters.  Some of the most common characters used, and their ANSI values are: 35 # (pound sign) 42 * (asterisk) 44 , (comma) 45 - (hyphen) 46 . (period) 47 / (forward slash) 60 < (less than) 62 > (greater than) 95 _ (underscore) 126 ~ (tilde) 183 · (bullet) 187 » (double bracket)
<b>.TrailLeaderSpacing</b>	Specifies the numeric variable that is assigned the number of spaces between leader characters. The maximum value is 9.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM TRAILSTATUS AS BOOLEAN
.FormatParaTabGet TRAILSTATUS, TLC&, TLS&
```

In the above example, the function assigns the active paragraph tag's trailing leader properties to **TRAILSTATUS**, **TLC**, and **TLS**, respectively.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"
.FormatParaTabGet TRAILSTATUS, TLC&, TLS&
.FormatParaTagEnd
```

---

{button ,AL("VENTURA\_FormatParaTabGet\_Menu;vent\_para;;;0,"Defaultoverview"),} [Related Topics](#)

## .FormatParaTabGetAt (VENTURA)

**.FormatParaTabGetAt** *.Position=long* , *.Location=single* , *.Align=long* , *.Leader=Boolean* , *.LeaderCharacter=long* , *.LeaderSpacing=long*

This function returns the attributes of a specified tab for either the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the [.FormatParaTagBegin](#) and [.FormatParaTagEnd](#) commands.

Syntax	Description
<b>.Position</b>	Specifies the position of the tab setting in the paragraph. Tab settings are ordered based on their position value in inches. The smallest value is the first ordered tab. The first tab setting is 1, the second tab setting is 2, and so on.
<b>.Location</b>	Specifies a numeric variable that is assigned the position of the specified tab in inches.
<b>.Align</b>	Specifies a numeric variable that is assigned the alignment of the specified tab: 0 Left (default if omitted) 1 Right 2 Decimal 3 Center
<b>.Leader</b>	Specifies a variable that is assigned the leader character status of the specified tab: TRUE (-1) if leader characters are enabled; otherwise, FALSE (0).
<b>.LeaderCharacter</b>	Specifies the numeric variable that is assigned the leader character as an ANSI character value. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters.  Some of the most common characters used and their ANSI values are: 35 # (pound sign) 42 * (asterisk) 44 , (comma) 45 - (hyphen) 46 . (period) 47 / (forward slash) 60 < (less than) 62 > (greater than) 95 _ (underscore) 126 ~ (tilde) 183 · (bullet) 187 » (double bracket)
<b>.LeaderSpacing</b>	Specifies the numeric variable that is assigned the number of spaces between leader characters. The maximum value is 9.

### Note



This command cannot be recorded.



The [.FormatParaTabCount](#) function can be used to count tab settings.

### Example

```
TabCount& = .FormatParaTabCount ( ) 'counts the numbers of tabs in the paragraph
DIM TabLead AS BOOLEAN 'declares a Boolean variable
.FormatParaTabGetAt TabCount, Locate!, TabAlign&, TabLead, Lchar&, Lspace&
```

The above example first counts the numbers of tabs in the active paragraph and returns this value to the **TabCount** variable. The properties of the last tab are then returned to the variables **Locate**, **TabAlign**, **TabLead**, **Lchar**, **Lspace** variables.

---

{button ,AL("VENTURA\_FormatParaTabGet\_Menu;vent\_para;;;",0,"Defaultoverview",)} [Related Topics](#)

## **.FormatParaTabSet (VENTURA)**

**.FormatParaTabSet** .Location=*single*, .Align=*long*, .Leader=*Boolean*, .LeaderCharacter=*long*, .LeaderSpacing=*long*


This command sets and modifies the tab settings for the active paragraph or changes the attributes of a paragraph tag.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
.Location	Specifies a new tab position or the position of an existing tab. The position is specified in inches.
.Align	Specifies the alignment of the new tab: 0 Left (default if omitted) 1 Right 2 Decimal 3 Center
.Leader	Specifies whether to use a leader character to fill the space created by the tab. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.LeaderCharacter	Specifies the leader character. The character is specified using ANSI character values. See the <a href="#">Corel SCRIPT character map</a> for more information about using ANSI characters. Some of the most common characters used and their ANSI values are: 35 # (pound sign) 42 * (asterisk) 44 , (comma) 45 - (hyphen) 46 . (period) 47 / (forward slash) 60 < (less than) 62 > (greater than) 95 _ (underscore) 126 ~ (tilde) 183 · (bullet) 187 » (double bracket) This parameter is used only if the <b>.Leader</b> parameter is enabled.
.LeaderSpacing	Specifies the number of spaces between leader characters. The maximum value is 9.

### **Note**

 This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

 To delete some or all tabs, use the **.FormatParaTabClear** command.

### **Example**

```
.FormatObjectBegin 1
.FormatParaTabSet 1.5, 2, -1, 183, 2
.FormatObjectEnd
```

The above example sets a tab at 1.5 inches using decimal alignment. The bullet character is used as a leader and there are two spaces between each bullet.

---

{button ,AL('VENTURA\_FormatParaTabSet\_Menu;vent\_para;;;',0,"Defaultoverview",)} [Related Topics](#)

## .FormatParaTagBegin (VENTURA)












**.FormatParaTagBegin** .TagName=string, .TagsInSelection=Boolean

Together, the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands are required to change a paragraph tag's attributes. The two commands enclose VENTURA commands that can modify a paragraph tag. The **.FormatParaTagBegin** statement is placed before the first paragraph modifying command and the **.FormatParaTagEnd** command is placed after the last paragraph modifying command.

Syntax	Description
.TagName	Specifies the paragraph tag to modify. The tag must already exist in the document. This parameter is required if the <b>TagsInSelection</b> parameter is omitted.  Multiple tags can be specified by separating them with semicolons. For example, "Tag1; Tag2; Tag3".
.TagsInSelection	Specifies whether to modify the tags in the current selection. Set to TRUE (-1) to enable this option; otherwise FALSE (0) which is the default if omitted. This parameter is mutually exclusive with the <b>TagName</b> parameter and takes precedence when set to TRUE. This parameter was introduced in Corel VENTURA 8.

### Note

The following commands can be used to modify a paragraph tag.

	<b><u>FormatParaAlignment command</u></b>
	<b><u>FormatParaBreaks command</u></b>
	<b><u>FormatParaBullet command</u></b>
	<b><u>FormatParaColor command</u></b>
	<b><u>FormatParaDefaults command</u></b>
	<b><u>FormatParaDropCap command</u></b>
	<b><u>FormatParaEffectColor command</u></b>
	<b><u>FormatParaEffectFont command</u></b>
	<b><u>FormatParaEffects command</u></b>
	<b><u>FormatParaFont command</u></b>
	<b><u>FormatParaHyphenation command</u></b>
	<b><u>FormatParaSpacing command</u></b>
	<b><u>FormatParaTab command</u></b>
	<b><u>FormatParaTabClear command</u></b>
	<b><u>FormatParaTabSet command</u></b>
	<b><u>FormatParaTypography command</u></b>

### Example

```
.FormatParaTagBegin "NewTag"  
  .FormatParaEffects .....  
  .FormatParaAlignment .....  
  .FormatParaDefaults .....  
.FormatParaTagEnd
```

In the above example, the **NewTag** paragraph tag has its Effects, Alignment, and Default attributes modified.

---

{button ,AL('VENTURA\_FormatParaBegin\_Menu;vent\_para;;;',0,"Defaultoverview",)} [Related Topics](#)



















## **.FormatParaTagEnd (VENTURA)**

### **.FormatParaTagEnd**

Together, the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands are required to change a paragraph tag's attributes. The two commands enclose VENTURA commands that can modify a paragraph tag. The **.FormatParaTagBegin** statement is placed before the first paragraph modifying command and the **.FormatParaTagEnd** command is placed after the last paragraph modifying command.

#### **Note**

The following commands can be used to modify a paragraph tag.

	<b><u>FormatParaAlignment command</u></b>
	<b><u>FormatParaBreaks command</u></b>
	<b><u>FormatParaBullet command</u></b>
	<b><u>FormatParaColor command</u></b>
	<b><u>FormatParaDefaults command</u></b>
	<b><u>FormatParaDropCap command</u></b>
	<b><u>FormatParaEffectColor command</u></b>
	<b><u>FormatParaEffectFont command</u></b>
	<b><u>FormatParaEffects command</u></b>
	<b><u>FormatParaFont command</u></b>
	<b><u>FormatParaHyphenation command</u></b>
	<b><u>FormatParaSpacing command</u></b>
	<b><u>FormatParaTab command</u></b>
	<b><u>FormatParaTabClear command</u></b>
	<b><u>FormatParaTabSet command</u></b>
	<b><u>FormatParaTypography command</u></b>

#### **Example**

```
.FormatParaTagBegin "NewTag"  
  .FormatParaEffects .....  
  .FormatParaAlignment .....  
  .FormatParaDefaults .....  
.FormatParaTagEnd
```

In the above example, the **NewTag** paragraph tag has its Effects, Alignment, and Default attributes modified.

---

{button ,AL('VENTURA\_FormatParaTagEnd\_Menu;vent\_para;;;',0,"Defaultoverview",,)} [Related Topics](#)

## **.FormatParaTypography (VENTURA)**

**.FormatParaTypography** *.WordSpace=long*, *.MinWordSpace=long*, *.MaxWordSpace=long*, *.LetterSpace=Boolean*, *.MaxLetterSpace=long*, *.VJAbove=long*, *.VJBelow=long*, *.VJInterLine=long*, *.Kerning=Boolean*

This command sets the typography attributes for the active paragraph or changes the attributes of a paragraph tag. The typography attributes set spacing between letters and lines, kerning, and sets limits on the amount of spacing added to a paragraph during vertical justification.

To change a paragraph tag's attributes, this command must be enclosed with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

To change the current (or selected) paragraph's attributes, this command must be enclosed with the **.FormatObjectBegin** and **.FormatObjectEnd** commands.

Syntax	Description
<code>.WordSpace</code>	Specifies the amount of space you want between words of unjustified text, as a percentage. A proportional value of 100 uses the default word spacing built into each font.
<code>.MinWordSpace</code>	Specifies the minimum amount of space allowed between words in justified paragraphs in tenths of a percent.
<code>.MaxWordSpace</code>	Specifies the maximum amount of space allowed between words in justified paragraphs in tenths of a percent.
<code>.LetterSpace</code>	Specifies whether to allow VENTURA to add space between letters as well as words when it justifies lines. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
<code>.MaxLetterSpace</code>	Specifies the maximum amount of space that VENTURA can add between letters in justified text when <b>.Letter Space</b> is enabled. This setting is specified in tenths of a percent of an em, a unit of measure roughly equal to the width of the capital M at the current font size.
<code>.VJAbove</code>	Specifies the maximum amount of space VENTURA can add above the paragraph as it vertically justifies columns, in tenths of a micron.
<code>.VJBelow</code>	Specifies the maximum amount of space VENTURA can add below the paragraph as it vertically justifies columns.
<code>.VJInterLine</code>	Specifies the maximum amount of space VENTURA can add between lines in the paragraph as it vertically justifies columns, in tenths of a micron.
<code>.Kerning</code>	Specifies whether to apply automatic pair kerning to the paragraph. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). For kerning to take effect, it must be enabled for the chapter, paragraph, and frame containing the paragraph.

### **Note**

**1** This command corresponds to the Paragraph Properties or Paragraph Tag Properties dialog box. In Corel VENTURA click Format, Paragraph Properties or click Format, Paragraph Tag Properties.

### **Example**

```
M_POINT = LENGTHCONVERT (1 , 3 , 1)
.FormatObjectBegin 1
.FormatParaTypography 115, , , , 6*M_POINT, 1*M_POINT
.FormatObjectEnd
```

The above example sets the normal word spacing to 115%, and the vertical adjustment maximum above and below the paragraph to 6 and 1 points, respectively.

The **LENGTHCONVERT** function creates a variable (**M\_POINT**) that is equal to the number of tenths of a micron in an point.

---

{button ,AL('VENTURA\_FormatParaTypography\_Menu;vent\_para;;;',0,"Defaultoverview"),} **Related Topics**

## .FormatParaTypographyGet (VENTURA)

**.FormatParaTypographyGet** .WordSpace=*long*, .MinWordSpace=*long*, .MaxWordSpace=*long*, .LetterSpace=*Boolean*, .MaxLetterSpace=*long*, .VJAbove=*long*, .VJBelow=*long*, .VJInterLine=*long*, .Kerning=*Boolean*

This function returns the typography attributes of the active paragraph's tag or for a specified paragraph tag. To specify the paragraph tag, enclose the function with the **.FormatParaTagBegin** and **.FormatParaTagEnd** commands.

Syntax	Description
.WordSpace	Specifies the numeric variable that is assigned the amount of space you want between words of unjustified text, as a percentage.
.MinWordSpace	Specifies the numeric variable that is assigned the minimum amount of space allowed between words in justified paragraphs in tenths of a percent.
.MaxWordSpace	Specifies the numeric variable that is assigned the maximum amount of space allowed between words in justified paragraphs in tenths of a percent.
.LetterSpace	Specifies the numeric variable that is assigned a value corresponding to whether the paragraph adds space between letters as well as words when it justifies lines: TRUE (-1) if this option is enabled; otherwise FALSE (0).
.MaxLetterSpace	Specifies the numeric variable that is assigned the maximum amount of space that VENTURA can add between letters in justified text when <b>.Letter Space</b> is enabled. This setting is specified in tenths of a percent of an em, a unit of measure roughly equal to the width of the capital M at the current font size.
.VJAbove	Specifies the numeric variable that is assigned the maximum amount of space VENTURA can add above the paragraph as it vertically justifies columns, in tenths of a micron.
.VJBelow	Specifies the numeric variable that is assigned the maximum amount of space VENTURA can add below the paragraph as it vertically justifies columns.
.VJInterLine	Specifies the numeric variable that is assigned the maximum amount of space VENTURA can add between lines in the paragraph as it vertically justifies columns, in tenths of a micron.
.Kerning	Specifies the numeric variable that is assigned a value corresponding to whether the paragraph applies automatic pair kerning: TRUE (-1) if this option is enabled; otherwise FALSE (0).

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatParaTypographyGet PercentSpace&, , , , , AboveSpace&
```

In the above example the function assigns the active paragraph tag's space between words attribute and the maximum space to add above a paragraph attribute to the variables **PercentSpace** and **AboveSpace**, respectively.

To use this example to return a specified paragraph tag's attributes (the Sample paragraph tag), use the following syntax:

```
.FormatParaTagBegin "Sample"  
  .FormatParaTypographyGet PercentSpace&, , , , , AboveSpace&  
.FormatParaTagEnd
```

---

{button ,AL("VENTURA\_FormatParaTypographyGet\_Menu;vent\_para;;;",0,"Defaultoverview",)} Related Topics

## **.FormatSetParaTag (VENTURA)**

**.FormatSetParaTag** .TagName=*string*

This command applies a specified paragraph tag to the active paragraph.

Syntax	Description
.TagName	Specifies the paragraph tag to apply.

### **Example**

```
.FormatSetParaTag "introduction"
```

The above example applies the "introduction" paragraph tag to the active paragraph.

---

{button ,AL('VENTURA\_ParaTagAddNew\_Menu;vent\_para;;;','0',"Defaultoverview",,)} [Related Topics](#)

## .ParaTagAddNew (VENTURA)

**.ParaTagAddNew .TagName=*string*, .CopyFrom=*string*, .TagType=*long*, .NextTagName=*string***

This command adds a new paragraph tag to the current style sheet. The new tag is based on an existing paragraph.

Syntax	Description
<b>.TagName</b>	Specifies the name of the new border tag. The tag name must not already exist in the style sheet.
<b>.CopyFrom</b>	Specifies the border tag that the new border tag is based on.
<b>.TagType</b>	Specifies the new tag's type: 0     Undefined 1     Body Text 2     Heading 3     List Item 4     Tabbed Text
<b>.NextTagName</b>	Specifies the paragraph tag that automatically follows the new tag when ENTER is pressed. This tag must exist in the current style sheet.

### Note



This command corresponds to the Tag Window. Click Tools, Tag Window.

### Example

```
.ParaTagAddNew "Intro", "BodyText", 2, "Heading3"
```

The above example adds a paragraph tag named Intro which is based on Body Text to the current style sheet. The new tag is a Heading type and is automatically followed by the Heading 3 style.

---

{button ,AL('VENTURA\_ParaTagAddNew\_Menu;vent\_para;;;',0,"Defaultoverview",)} [Related Topics](#)


# .ParaTagCopy (VENTURA)

**.ParaTagCopy** .TagName=*string*

This command copies a paragraph tag from the current style sheet to the clipboard. This command can be used to copy tags into VENTURA Libraries.

Syntax	Description
.TagName	Specifies the name of the paragraph tag to copy.

**Note**

 This command corresponds to the Tag Window. Click Tools, Tag Window.

**Example**

```
.ParagraphTagCopy "TradeMark"  
.FileLibraryNew "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste
```

The above example copies the paragraph tag TradeMark, creates a VENTURA library named MYLIB.VLB, and pastes the TradeMark tag into the new library.


## .ParaTagCount (VENTURA)

**ReturnValue**& = .ParaTagCount ( )

This function returns the number of paragraph tags in the current style sheet.

Syntax	Description
<b>ReturnValue</b> &	Specifies the numeric variable that is assigned the number of paragraph tags in the current style sheet.

### Note

 This command cannot be recorded.

### Example

```
Pcounts = .ParaTagCount ( )
```


The number of paragraph tags is assigned the **Pcounts** variable.


---

{button ,AL("VENTURA\_ParaTagCount\_Menu;vent\_para;;;','0,"Defaultoverview"),} [Related Topics](#)

## .ParaTagDelete (VENTURA)


**.ParaTagDelete** *.TagName=string*, *.ReformatTagName=string*


This command deletes a paragraph tag from the current style sheet. Any item in the document  and in other documents that are linked to the same style sheet through the Library

 formatted with the deleted tag will be reformatted using another tag of your choice.

Syntax	Description
<b>.TagName</b>	Specifies the name of the paragraph tag to delete.
<b>.ReformatTagName</b>	Specifies the paragraph tag to apply to paragraphs tagged with <b>.TagName</b> .

### Note

 This command corresponds to the Tag Window. Click Tools, Tag Window.

 You cannot delete the Body Text paragraph tag.

### Example

`.ParaTagDelete "Intro" , "Body Text"`

The above example deletes the paragraph tag Intro and applies the Body Text tag to paragraphs using the Body Text tag.

---

`{button ,AL('VENTURA_ParaTagDelete_Menu;vent_para;;;',0,"Defaultoverview"),}` [Related Topics](#)




## .ParaTagGetAt (VENTURA)

ReturnString\$ = .ParaTagGetAt (.TagIndex=*long*)

This function returns the paragraph tag name associated with a paragraph tag index number in the current style sheet.

Syntax	Description
ReturnString\$	Specifies the string variable that is assigned the name of a paragraph tag.
.TagIndex	Specifies the index number of a paragraph tag. Index numbers are not associated with tag order in the Tag Window. Instead, index numbers are associated by the order in which the paragraph tags are created. The first tag created is 1, the second tag created is 2, and so on. The last created tag is equal to the return value from the <a href="#">.ParaTagCount</a> function. If a tag is deleted, the tag index is recompiled.

### Note

 This command cannot be recorded.

### Example

```
LastTag& = .ParaTagCount ( )      'number of tags  
LastTagName$ = .ParaTagGetAt (LastTag&)
```

In the above example, the first statement counts the number of paragraph tags. The second statement is assigned the name of last created paragraph tag still available in the current style sheet.

---

{button ,AL("VENTURA\_ParaTagGetAt\_Menu;vent\_para;;;","0,"Defaultoverview",)} [Related Topics](#)

# .ParaTagRename (VENTURA)

**.ParaTagRename .OldTagName=*string*, .NewTagName=*string***

This command renames a specified paragraph tag. Renaming tags may affect the formatting of other documents that are linked to the same style sheet through the Library.

Syntax	Description
.OldTagName	Specifies the paragraph tag to be renamed.
.NewTagName	Specifies the paragraph new tag name.

**Note**



This command corresponds to the Tag Window. Click Tools, Tag Window.



You cannot rename any generated tags (those with a Z\_ prefix).

**Example**

`.ParaTagRename "Name 1", "Name 2"`

The above example renames the paragraph tag Name 1 to Name 2.

---

`{button ,AL('VENTURA_ParaTagRename_Menu;vent_para;;;','0,"Defaultoverview",,)} Related Topics`

## **.TextArrangeSideBySide (VENTURA)**

### **.TextArrangeSideBySide**

This command places the selected paragraphs side by side.

#### **Note**



To use this command, the paragraphs must be selected and contiguous.

#### **Example**

.TextArrangeSideBySide

This example places the selected paragraphs side by side.

---

{button ,AL('VENTURA\_ParaTagRename\_Menu;vent\_para;;;',0,"Defaultoverview",,)} Related Topics

## **.ConvertShapeToPicture (VENTURA)**

### **.ConvertShapeToPicture**

This command converts a graphic object in a embedded picture file.

#### **Note**



This command was introduced in Corel VENTURA 8.



This command corresponds to selecting a graphic object in Corel VENTURA and right-clicking and clicking Convert, Convert To Picture.

#### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.DrawEllipse 2*M_INCH, 2*M_INCH, 2.5*M_INCH, 4.3*M_INCH
.ConvertShapeToPicture
```

The above example draws an ellipse with a bounding box of 2.5 by 4.3 inches. The bounding box is 2 inches from the left and top of the base page frame. The graphic object is then converted to a picture.

---

{button ,AL('vent\_pictures;;;;','0',"Defaultoverview",)} [Related Topics](#)

## .CurrentPictureFile (VENTURA)

**ReturnString\$ = .CurrentPictureFile ( )**

This function returns the file name of the picture file in the selected frame.

Syntax	Description
<b>ReturnString\$</b>	Specifies a string variable that is assigned the name of the picture file. If the picture file is embedded, the file name that appears in the VENTURA Navigator or File List is assigned. If the picture file is external, picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.  This parameter was modified in Corel VENTURA 8.

### Note



This command cannot be recorded.



If more than one frame is selected, this function returns the name of the picture file in the first selected frame.

### Example

```
PicName$ = .CurrentPictureFile ( )
```

The above example assigns the name of the picture file in the active frame to the **PicName\$** string variable.


---

**{button ,AL('vent\_pictures;;;;','0',"Defaultoverview",)} Related Topics**

## .FileExportPicture (VENTURA)

**.FileExportPicture .FileName=string, .NewFileName=string, .FilterSectionName=string**

This command exports picture files from the current chapter. The picture files may be embedded or external.

Syntax	Description
<b>.FileName</b>	Specifies the name of the picture file to export. If the picture file is embedded, use the file name that appears in the VENTURA Navigator or File List. If the picture file is external, use the picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.  This parameter was modified in Corel VENTURA 8.
<b>.NewFileName</b>	Specifies the name and path of the exported picture file. The file extension used determines the filter type. Default filter settings are used with this command.
<b>.FilterSectionName</b>	Specifies a string corresponding to a filter type. Click  for a list a valid picture export filters. For example, Windows Bitmap uses the string "BMP".

### Note



This command was introduced in Corel VENTURA 8.



This command corresponds to the Export Picture command on the File menu in Corel VENTURA. Click File, Export Picture.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileImportPicture "c:\Windows\cars.bmp", 0, FALSE
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatAttachPicture "cars"
.FileExportPicture "cars", "c:\my_files\cars.cpt", "CPT"
```

The above example imports a picture file named CARS.BMP (a Windows bitmap) into the active chapter. The **.SelectObjectAt** command is used to select an existing frame with the imported picture file attached to it. The cars bitmap is then exported as a Corel PHOTO-PAINT file.


---


{button ,AL('vent\_pictures;;;;','0',"Defaultoverview"),} [Related Topics](#)

## .FileImportPicture (VENTURA)


**.FileImportPicture** **.FileName=string**, **.FilterType=long**, **.ReplaceExisting=Boolean**, **.External=Boolean**


This command imports pictures  illustrations, photographs, charts, etc.

 created with painting, illustration, drawing, charting and scanning programs.

Syntax	Description
<b>.FileName</b>	Specifies the name of the picture to open. If path information is not included, the active folder is used.
<b>.FilterType</b>	In Corel VENTURA 7, this parameter specified a value corresponding to a filter type. In Corel VENTURA 8, this parameter should be set to 0  Corel VENTURA determines and uses the appropriate filter. If omitted, it is set to 0.
<b>.ReplaceExisting</b>	Specifies whether to remove the currently attached picture file from the chapter and replace with the specified picture file. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0) which is the default.
<b>.External</b>	<p>Specifies whether to incorporate the picture file into the active document rather than storing a reference to it. Set to TRUE (-1) to reference the picture file. Set to FALSE (0) to embed the picture file into the active VENTURA document. If this parameter is omitted it is set to FALSE unless the picture file size exceeds the limit set in the File Imports warning option control in VENTURA (click Tools, Options, Save tab).</p> <p>When a picture file is referenced, VENTURA retrieves and displays it whenever the document is opened or the page it appears on is displayed. Referencing picture files helps to keep the file size of the document to a minimum.</p>

### Note

 If a frame is selected before this command is issued, the imported picture is placed in the selected frame. If a frame isn't selected before this command is issued, the imported picture can be attached to a frame with the **.FormatAttachPicture** command.

 This command corresponds to the Import Picture command on the File menu in Corel VENTURA. Click File, Import Picture.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileImportPicture "c:\Windows\cars.bmp", 0, FALSE
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatAttachPicture "cars.bmp"
```

The above example imports a picture file named CARS.BMP (a Windows bitmap) into the active chapter. The **.SelectObjectAt** command is used to select an existing frame with the imported picture file attached to it.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---


{button ,AL('vent\_pictures;;;;','0',"Defaultoverview"),} **Related Topics**


## .FileImportPictureCrop (VENTURA)

**.FileImportPictureCrop .FileName=string, .FilterType=long, .Left=long, .Top=long, .Width=long, .Height=long**


This command is obsolete in Corel VENTURA 8.

This command imports pictures  illustrations, photographs, charts, etc.

 created with painting, illustration, drawing, charting and scanning programs. This command can also crop a picture before importing it.

Syntax	Description
<b>.FileName</b>	Specifies the name of the document to open. If path information is not included, the active folder is used.
<b>.FilterType</b>	In Corel VENTURA 7, this parameter specified a value corresponding to a filter type. In Corel VENTURA 8, this parameter should be set to 0  Corel VENTURA determines and uses the appropriate filter.
<b>.Left</b>	Specifies the number of pixels to be cropped from the left side of the imported picture. If omitted, the default is 0.
<b>.Top</b>	Specifies the number of pixels to be cropped from the top of the imported picture. If omitted, the default is 0.
<b>.Width</b>	Specifies the width of the picture in pixels. If omitted, the default is the original picture's width less the number of pixels specified with the <b>.Left</b> parameter.
<b>.Height</b>	Specifies the height of the picture in pixels. If omitted, the default is the original picture's height less the number of pixels specified with the <b>.Top</b> parameter.

### Note

 This command corresponds to the Import Picture command on the File menu in Corel VENTURA. Click File, Import Picture.

### Example

```
.FileImportPictureCrop "c:\Windows\cars.bmp", 0, 2, 4, 25, 27
```

The above example imports a picture file named CARS.BMP (a Windows bitmap) into the current publication. The left-side of the picture is cropped by 2 pixels, the top-side by 4 pixels. The width and height of the picture is set to 25 and 27.

---

{button ,AL(vent\_pictures;;;;;'0,"Defaultoverview",)} Related Topics





## .FileImportPictureResample (VENTURA)

**.FileImportPictureResample .FileName=string, .FilterType=long, .Width=long, .Height=long, .Horizontal=long, .Vertical=long**


This command is obsolete in Corel VENTURA 8.

This command imports pictures  illustrations, photographs, charts, etc.

 created with painting, illustration, drawing, charting and scanning programs. This command can also resample a picture before importing it.

Syntax	Description
<b>.FileName</b>	Specifies the name of the document to open. If path information is not included, the active folder is used.
<b>.FilterType</b>	In Corel VENTURA 7, this parameter specified a value corresponding to a filter type. In Corel VENTURA 8, this parameter should be set to 0  Corel VENTURA determines and uses the appropriate filter.
<b>.Width</b>	Specifies the width of the picture in pixels.
<b>.Height</b>	Specifies the height of the picture in pixels.
<b>.Horizontal</b>	Specifies horizontal resolution of the imported picture in dots per inch.
<b>.Vertical</b>	Specifies vertical resolution of the imported picture in dots per inch.

### Note

 This command corresponds to the Import Picture command on the File menu in Corel VENTURA. Click File, Import Picture.

### Example

```
.FileImportPictureResample "c:\Windows\cars.bmp", 0, 64, 64, 72, 100
```

The above example imports a picture file named CARS.BMP (a Windows bitmap) into the current publication. The imported picture is resized to 64 pixels by 64 pixels. The horizontal and vertical resolutions are set to 72 and 100 dots per inch.

---

{button ,AL(vent\_pictures;;;;;'0,"Defaultoverview",)} Related Topics




## **.FormatAttachPicture (VENTURA)**

**.FormatAttachPicture .FileName=string**

This command attaches a specified picture file to the currently selected frame. The picture file must already exist in the chapter before it can be attached.

Syntax	Description
<b>.FileName</b>	Specifies the name of the picture file to attach. If the picture file is embedded, use the file name that appears in the VENTURA Navigator or File List. If the picture file is external, use the picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.  This parameter was modified in Corel VENTURA 8.

### **Note**

-  You can detach a picture file from a frame using the [.FormatDetachPicture](#) command.
-  If a picture already exists in the selected frame when the **.FormatAttachPicture** command is issued, the existing picture is hidden by the newly attached picture.
-  You can attach a text file using the [.FormatAttachTextFile](#) command.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileImportPicture "c:\Windows\cars.bmp", 0, 0
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatAttachPicture "cars"
```

The above example imports a picture file named CARS.BMP (a Windows bitmap) into the active chapter. The [.SelectObjectAt](#) command is used to select an existing frame, and the imported picture file is attached to it.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

**{button ,AL('vent\_pictures;;;;','0',"Defaultoverview",)} [Related Topics](#)**

## **.FormatDetachPicture (VENTURA)**

**.FormatDetachPicture .FileName=string**

This command removes a specified picture file from the currently selected frame and chapter.

Syntax	Description
<b>.FileName</b>	<p>Specifies the name of the picture file to detach. If the picture file is embedded, use the file name that appears in the VENTURA Navigator or File List. If the picture file is external, use the picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.</p> <p>This parameter was modified in Corel VENTURA 8.</p> <p>If not specified, the picture file in the active frame is detached.</p>

### **Note**



You can attach a picture file to a frame using the [.FormatAttachPicture](#) command.



You can detach a text file using the [.FormatDetachTextFile](#) command.

### **Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatDetachPicture "c:\Windows\cars.bmp"
```

The above example selects an existing frame and removes an imported Windows bitmap from it.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

`{button ,AL('vent_pictures;;;;','0','Defaultoverview'),}` [Related Topics](#)




# .PictureFileCopy (VENTURA)

**.PictureFileCopy .FileName=string**

Copies a specified picture file in the active chapter to the clipboard.

Syntax	Description
<b>.FileName</b>	Specifies the name of the picture file to copy. If the picture file is embedded, use the file name that appears in the VENTURA Navigator or File List. If the picture file is external, use the picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.  This parameter was modified in Corel VENTURA 8.

**Note**

-  This command corresponds to selecting a picture file in the VENTURA Navigator, right-clicking and selecting Copy.
-  The clipboard contents can be pasted into a VENTURA library, a frame, or another chapter.
-  Use the **.EditCopy** command to copy a selection.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileImportPicture "c:\Windows\cars.bmp", 0, FALSE
.PictureFileCopy "cars"
```

The above example imports a picture file named CARS.BMP (a Windows bitmap) into the active chapter. The **.SelectObjectAt** command is used to select an already existing frame, and the imported picture file is attached to it. The picture is then copied to the clipboard.

The **.LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

```
.FileImportPicture "c:\Windows\cars.bmp", 0
.PictureFileCopy "cars"
```

The above example imports a picture file named CARS.BMP (a Windows bitmap) into the current publication. The picture is then copied to the clipboard.

## **.PictureFileCount (VENTURA)**

**ReturnValue**& = .PictureFileCount ( )

This function returns the number of picture files in the active chapter.

Syntax	Description
<b>ReturnValue</b> &	Specifies the numeric variable that is assigned the number of picture files in the active chapter.

### **Note**

**I** This command cannot be recorded.

### **Example**

```
PicCount = .PictureFileCount ( )
```

In the above example, the number of picture files is assigned to the **PicCount** variable.

---

{button ,AL('vent\_pictures;;;;','0',"Defaultoverview"),} [Related Topics](#)


## .PictureFileEmbed (VENTURA)

**.PictureFileEmbed** *.FileName=string*

This command embeds an external picture file.

Syntax	Description
<b>.FileName</b>	Specifies the name of the picture file to embed. Use the picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.

### Note

 This command was introduced in Corel VENTURA 8.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileImportPicture "c:\Windows\cars.bmp", 0, FALSE (0), -1
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatAttachPicture "c:\Windows\cars.bmp"
.FileEmbed "c:\Windows\cars.bmp"
```

The above example imports a picture file named CARS.BMP (a Windows bitmap) into the active chapter. The **.SelectObjectAt** command is used to select an existing frame with the imported picture file attached to it. The cars bitmap embedded.

---

{button ,AL('vent\_pictures;;;;','0',"Defaultoverview"),} Related Topics

## .PictureFileGetAt (VENTURA)

**ReturnString\$ = .PictureFileGetAt .FileIndex=long**

This function returns the name of specified picture file in the active Corel VENTURA chapter.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of the specified picture file. If the picture file is embedded, the file name that appears in the VENTURA Navigator or File List is assigned. If the picture file is external, picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.  This parameter was modified in Corel VENTURA 8.
<b>.FileIndex</b>	Specifies the picture file as an index number. The first picture file corresponds to 1, the second picture file corresponds to 2, and so on. The index numbers are based on the order in which the files are inserted into the chapter. If a picture file is deleted from the chapter, the index numbers are re-ordered.

### Note

 This command cannot be recorded.

### Example

```
PictureName = .PictureFileGetAt (7)
```

In the above example, the name of index picture file 7 is assigned to the variable **PictureName**.

```
NumberOfPictureFiles% = . PictureFileCount ( )  
FOR i% = 1 TO NumberOfTextPicture%  
    PictureFileName$(i%) = . PictureFileGetAt (i%)  
NEXT i%
```

In this example, the number of picture files in the active chapter are used in a FOR...NEXT LOOP to create a string array of all the text files in the active chapter.

---

**{button ,AL(^vent\_pictures;;;;;'0,"Defaultoverview",)} [Related Topics](#)**

## .PictureFileIsLinked (VENTURA)

**ReturnNumber** = .PictureFileIsLinked .FileName=*string*

This function returns a value indicating whether a picture file is embedded or external (linked).

Syntax	Description
<b>ReturnNumber</b>	Specifies the numeric variable that is assigned a value corresponding to whether a specified picture file is embedded or external: TRUE (-1) if external (linked); otherwise FALSE (0).
<b>.FileName</b>	Specifies the name of the picture file to test. If the picture file is embedded, use the file name that appears in the VENTURA Navigator or File List. If the picture file is external, use the picture file's full name and path. This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.  This parameter was modified in Corel VENTURA 8.

### Note



This function cannot be recorded.



This function was introduced in Corel VENTURA 8.

### Example

```
PicStatus = .PictureFileIsLinked (cars)
```

The above example assigns a value to **PicStatus** indicating whether the **cars** picture file is embedded or external.

---

**{button ,AL('vent\_pictures;;;;','0',"Defaultoverview",)} Related Topics**



## .PictureFileIsUsed (VENTURA)

**ReturnString** = .PictureFileIsUsed **.FileName**=*string*

This function returns a value indicating whether a specified picture file in the active chapter is contained within a frame.

Syntax	Description
<b>ReturnString</b>	Specifies a numeric variable that is assigned a value indicating whether a specified picture file is contained within a frame. If contained TRUE (-1); otherwise FALSE (0).
<b>.FileName</b>	Specifies the name of the picture file to test. If the picture file is embedded, use the file name that appears in the VENTURA Navigator or File List. If the picture file is external, use the picture file's full name and extension (e.g., abc.bmp). This information is available by right-clicking the external picture file in the Corel VENTURA Navigator, clicking Picture File Properties, and clicking the General tab.  This parameter was modified in Corel VENTURA 8.

### Example

```
PictureFiles& = .PictureFileCount( )
FOR i% = 1 TO PictureFiles&
    PictureName$ = .PictureFileGetAt (i%)
    InUseStatus = .PictureFileIsUsed (PictureName$)

    IF InUseStatus = TRUE THEN MESSAGE "This picture file is used"
    IF InUseStatus = FALSE THEN MESSAGE "This picture file is not used"
NEXT i%
```

The above example tests whether the picture files in the active chapter are contained within a frame. The first line assigns the number of picture files in the active chapter to the **PictureFiles** variable. This variable is then used to set the maximum looping value. The **i%** variable is assigned to the **.PictureFileGetAt** function as an index number. This function returns the name each picture file in the active chapter. Each picture file is then tested for frame containment.

---

{button ,AL('vent\_pictures;;;;','0',"Defaultoverview",)} [Related Topics](#)

## .CurrentPageNumber (VENTURA)

**.CurrentPageNumber** .LeftPage=*Boolean* , .NumberType=*long*

This function returns the current page number.

Syntax	Description
.LeftPage	Specifies an optional numeric variable which is assigned a value indicating whether the current page is Left or Right. If Left, TRUE (-1) is assigned; if Right, FALSE (0) is assigned.
.NumberType	Specifies which page number to retrieve: 1 User page number (default if omitted) 2 Absolute page number, relative to publication 3 Page counter, relative to current chapter This parameter was added in Corel VENTURA 8.

### Example

```
PageVal& = .CurrentPageNumber PageSide, 2
```

This example returns the current page number to **PageVal** and the Left/Right value to **PageSide**. The page number is absolute.

---

{button ,AL('VENTURA\_Page\_Menu;;;','0,"Defaultoverview",)} [Related Topics](#)

# .PageDeletePages (VENTURA)

**.PageDeletePages** *.StartPage=long, .EndPage=long, .ChapterName=string*

This command deletes pages inserted using the Insert Pages command in the Insert menu or the **.PageInsertPages** command. Text and pictures on inserted pages remain in the Files list; however, frames and graphic objects are deleted with the page.

Syntax	Description
<b>.StartPage</b>	Specifies the number of the page to delete. The number specified refers to the location of the page in the chapter not the whole document. So, although the page number <b>I</b> as it appears in the header or footer
<b>I</b>	of the fourth page in the second chapter may be 23, to delete it, enter the number 4 in the Delete Page box. If not specified, the current page is deleted.
<b>.EndPage</b>	Specifies the number of the last page in the range of pages to delete. If not specified, the current page is deleted.
<b>.ChapterName</b>	Specifies the chapter from which to delete page. If not specified, the pages are removed from the active chapter.

## Note

- I** This command corresponds to the Delete Pages dialog box in Corel VENTURA. Click Page, Delete Pages.
- I** You cannot use this command to remove pages VENTURA creates automatically. Such pages are removed for you when you delete the text, frames, or other items on them.

## Example

`.PageDeletePages 2, 5, "Intro"`

The above example removes pages 2-5 in the Intro chapter.

---

`{button ,AL('VENTURA_Page_Menu;;;',0,"Defaultoverview"),}` [Related Topics](#)

## **.PageEnableFooter (VENTURA)**

**.PageEnableFooter** *.Enable=Boolean*

This command enables the footer. This command can be applied to the current page or the base page.

Syntax	Description
<b>.Enable</b>	Set to TRUE (-1) to enable the footer. Set to FALSE (0) to disable the footer. Toggles if omitted.

### **Example**

```
.PageEnableFooter TRUE
```

This example enables the footer on the current page.

---

{button ,AL('VENTURA\_Page\_Menu;;;','0',"Defaultoverview"),} [Related Topics](#)

## **.PageEnableFootnotes (VENTURA)**

**.PageEnableFootnotes .Enable=*Boolean***

This command enables the footnotes and endnotes. This command can be applied to the current page or the base page.

Syntax	Description
<b>.Enable</b>	Set to TRUE (-1) to enable the footnotes and endnotes. Set to FALSE (0) to disable. Toggles if omitted.

### **Example**

```
.PageEnableFootnotes TRUE
```

This example enables footnotes on the current page.

---

{button ,AL('VENTURA\_Page\_Menu;;;','0',"Defaultoverview"),} [Related Topics](#)

## **.PageEnableHeader (VENTURA)**

**.PageEnableHeader .Enable=*Boolean***

This command enables the header. This command can be applied to the current page or the base page.

Syntax	Description
<b>.Enable</b>	Set to TRUE (-1) to enable the header. Set to FALSE (0) to disable the header. Toggles if omitted.

### **Examples**

```
.PageEnableHeader TRUE
```

This example enables the header on the current page.

---

```
{button ,AL('VENTURA_Page_Menu;;;','0',"Defaultoverview"),}
```

[Related Topics](#)

## **.PageFacingPageProperties (VENTURA)**

**.PageFacingPageProperties** *.Mirror=Boolean*

This command mirrors the properties from the facing page to the current page.

Syntax	Description
.Mirror	Set to TRUE (-1) to mirror the properties from the facing page. Otherwise, set to FALSE (0) which is the default if omitted.

### **Note**

**I** This command is not recordable.

### **Example**

```
.ViewZoomLayout 3, 1
.ViewMasterPage
.PageFacingPageProperties TRUE
```

This example mirrors the properties from the facing page.

---

{button ,AL('VENTURA\_Page\_Menu;;;',0,"Defaultoverview"),} [Related Topics](#)

**.PageGrid (VENTURA)**

**.PageGrid** .HorzFrequency=*long*, .VertFrequency=*long*, .LocationX=*long*, .LocationY=*long*, .Show=*Boolean*

This command sets the grid settings for Corel VENTURA.

Syntax	Description
.HorzFrequency	Specifies the spacing of the horizontal grid lines in tenths of a micron.
.VertFrequency	Specifies the spacing of the vertical grid lines in tenths of a micron.
.LocationX	Specifies the horizontal starting point for the grid relative to the left side of the <u>base page frame</u> , in tenths of micron.
.LocationY	Specifies the vertical starting point for the grid relative to the top of the <u>base page frame</u> , in tenths of micron.
.Show	Specifies whether to show the grid. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). FALSE is the default if omitted.

**Note**

**I** This command corresponds to the Grid tab on the Grid & Guideline Properties dialog box in Corel VENTURA. Click Tools, Grid & Guideline Properties, Grid tab.

**I** This command is not recordable.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.PageGrid 2*M_INCH, 1*M_INCH, 0.5*M_INCH, 0.75*M_INCH, TRUE
```

The above example enables the grid in VENTURA. The horizontal lines are spaced 2 inches apart and the vertical lines are spaced 1 inch apart. The starting point for the grid is 0.5 and 0.75 (X, Y) inches relative to the top-left corner of the base page frame.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_Page\_Menu;;;','0,"Defaultoverview",)} Related Topics



## .PageGridGet (VENTURA)

**.PageGridGet** .HorzFrequency=*long*, .VertFrequency=*long*, .LocationX=*long*, .LocationY=*long*, .Show=*Boolean*

This function returns Corel VENTURA grid settings.

Syntax	Description
.HorzFrequency	Specifies the numeric variable that is assigned the spacing of the horizontal grid lines in tenths of a micron.
.VertFrequency	Specifies the numeric variable that is assigned the spacing of the vertical grid lines in tenths of a micron.
.LocationX	Specifies the numeric variable that is assigned the horizontal starting point for the grid relative to the left side of the <u>base page frame</u> , in tenths of micron.
.LocationY	Specifies the numeric variable that is assigned the vertical starting point for the grid relative to the top of the <u>base page frame</u> , in tenths of micron.
.Show	Specifies the variable that is assigned a value corresponding to whether the grid is displayed: TRUE (-1) if displayed; FALSE (0) if not displayed.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
DIM GridSet AS BOOLEAN
.PageGridGet X&, Y&, XX&, YY&, GridSet
```

The above example assigns the grid settings to the variables **X&**, **Y&**, **XX&**, **YY&**, and **GridSet**.

---

{button ,AL("VENTURA\_Page\_Menu;;;','0,"Defaultoverview",)} [Related Topics](#)


## .PageGuideline (VENTURA)

**.PageGuideline** .Show=*Boolean*, .Lock=*Boolean*

This command sets guideline display and lock options.

Syntax	Description
.Show	Specifies whether to show guidelines. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
.Lock	Specifies whether to lock guidelines so that they cannot be moved with the mouse. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0), which is the default if omitted.

### Note

 This command corresponds to the Guidelines tab on the Grid & Guideline Properties dialog box in Corel VENTURA. Click Page, Grid/Guideline Setup, Guidelines tab.

### Example

```
.PageGuideline TRUE, TRUE
```

The above example displays previously set guidelines and locks them in place.

---

{button ,AL("VENTURA\_Page\_Menu;;;','0,"Defaultoverview"),} [Related Topics](#)


## .PageGuidelineAdd (VENTURA)

**.PageGuidelineAdd .Position=*long*, .Vertical=*Boolean***

This command adds a guideline to the active document.

Syntax	Description
<b>.Position</b>	Specifies the position of the guideline to remove, in tenths of a micron relative to <u>base page frame</u> .
<b>.Vertical</b>	Specifies whether the guideline to add (specified in the <b>.Position</b> parameter) is a vertical or horizontal guideline. Set to TRUE (-1) for vertical guidelines; otherwise, set to FALSE (0), which is the default if omitted.

### Note

 This command corresponds to the Guidelines tab on the Grid & Guideline Properties dialog box in Corel VENTURA. Click Page, Grid/Guideline Setup, Guidelines tab.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.PageGuidelineAdd 2*M_INCH, TRUE
```

The above example adds a vertical guideline that is 2 inches from the left side of the base page frame. The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_Page\_Menu;;;','0',"Defaultoverview",)} Related Topics

## .PageGuidelineCount (VENTURA)

**Return Value** & = .PageGuidelineCount .Vertical=*Boolean*

This function returns the number of vertical or horizontal guidelines in the active document.

Syntax	Description
<b>Return Value</b> &	Specifies the numeric variable that is assigned the number of guidelines.
<b>.Vertical</b>	Specifies whether to return the number of vertical or horizontal guidelines. Set to TRUE (-1) to return vertical guidelines; otherwise, set to FALSE (0) to return the number of horizontal guidelines, which is the default if omitted.

### Note



This command cannot be recorded.

### Example

```
VertGLCount& = .PageGuidelineCount (TRUE)
```

The above example assigns the number of vertical guidelines in the active document.

---

{button ,AL('VENTURA\_Page\_Menu;;;','0,"Defaultoverview",,)} [Related Topics](#)


## .PageGuidelineDelete (VENTURA)

**.PageGuidelineDelete** .Position=*long*, .Vertical=*Boolean*, .All=*Boolean*

This command removes a specified guideline or all the guidelines in the active document.

Syntax	Description
.Position	Specifies the position of the guideline to remove, in tenths of a micron relative to <u>base page frame</u> .
.Vertical	Specifies whether the guideline specified in the <b>.Position</b> parameter is a vertical or horizontal guideline. Set to TRUE (-1) for vertical guidelines; otherwise, set to FALSE (0), which is the default if omitted.
.All	Specifies whether to delete all the guidelines in the active document. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0), which is the default if omitted.

### Note

 This command corresponds to the Guidelines tab on the Grid & Guideline Properties dialog box in Corel VENTURA. Click Page, Grid/Guideline Setup, Guidelines tab.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.PageGuidelineDelete 2*M_INCH, TRUE
```

The above example removes the vertical guideline that is 2 inches from the left side of the base page frame. The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

```
.PageGuidelineDelete , , TRUE
```

The above example removes all the guidelines from the active document:

---

{button ,AL('VENTURA\_Page\_Menu;;;','0',"Defaultoverview",)} Related Topics

## **.PageGuidelineGet (VENTURA)**

**.PageGuidelineGet** *.Show=Boolean, .Lock=Boolean*

This function returns guideline display and lock options in the active document.

Syntax	Description
.Show	Specifies a numeric variable that is assigned a value indicating whether the show guidelines option is enabled: TRUE (-1) if this option is enabled; otherwise, FALSE (0).
.Lock	Specifies a numeric variable that is assigned a value indicating whether the lock guidelines option is enabled: TRUE (-1) if this option is enabled; otherwise, FALSE (0).

### **Note**



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.



Use the .PageGuideline command to set show and lock options.

### **Example**

```
DIM GuideShow AS BOOLEAN
DIM GuideLock AS BOOLEAN
.PageGuidelineGet GuideLock, GuideShow
```

The above example returns the active document's guideline show and lock options to the variables **GuideShow** and **GuideLock**, respectively.

---

{button ,AL('VENTURA\_Page\_Menu;;;','0',"Defaultoverview",)} Related Topics

## .PageGuidelineGetAt (VENTURA)

**.PageGuidelineGetAt** *.Index=long, .Position=long, .Vertical=Boolean*

This function returns the position and orientation of a specified guideline in the active document.

Syntax	Description
<b>.Index</b>	Specifies the index number of a guideline. Index numbers are associated by the order in which the guidelines are created. The first guideline created is 1, the second guideline created is 2, and so on. If a guideline is deleted, the index numbers are recompiled.
<b>.Position</b>	Specifies the numeric variable that is assigned the position of the specified guideline, in tenths of a micron relative to <u>base page frame</u> .
<b>.Vertical</b>	Specifies the variable that is assigned the orientation of the specified guideline: TRUE (-1) for vertical guidelines; FALSE (0) for horizontal guidelines.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
VertGLCount& = .PageGuidelineCount (TRUE) 'counts vertical guidelines
HorzGLCount& = .PageGuidelineCount (FALSE) 'counts horizontal guidelines
TotalGL& = VertGLCount& + HorzGLCount& 'total number of guidelines
```

The above portion of the example determines the total number of guidelines (**TotalGL&**) in the active document. This value is then used to determine the position and orientation of the last created guideline:

```
DIM OrientLast AS BOOLEAN
.PageGuidelineGetAt TotalGL&, PosLast&, OrientLast
```

---

{button ,AL('VENTURA\_Page\_Menu;;;','0',"Defaultoverview"),} Related Topics

## .PageInsertPages (VENTURA)

**.PageInsertPages** .ChapterName=*string*, .NumberOfPages=*long*, .AtPage=*long*, .Before=*Boolean*, .MasterPage=*string*

This command adds blank pages to the active document. Manually inserting pages is necessary for documents in which all text is placed in frames on top the base page rather than on it. VENTURA automatically creates additional pages to accommodate text on the base page. You might also insert a page at the beginning of a document to prevent text on the base page from flowing onto the cover page.

Syntax	Description
.ChapterName	Specifies the name of the chapter to insert pages into. If not specified, the active chapter is used.
.NumberOfPages	Specifies the number of pages to insert. If you are placing a text file on the inserted page, you only need to insert one page. Additional pages will be added automatically to accommodate the rest of the text file.
.AtPage	Specifies the page, before or after which, to insert the page(s). If not specified, the current page is used.
.Before	Specifies whether to insert the page(s) before or after the page specified with <b>.AtPage</b> . Set to TRUE (-1) to insert before. Set to FALSE to insert after, which is the default if omitted.
.MasterPage	Specifies the name of the page tag to apply to the newly inserted page(s). If not specified, the Default Page tag is used.

### Note



This command corresponds to the Insert Pages dialog box in Corel VENTURA. Click Page, Insert Pages.



VENTURA must be in text mode when this command is issued.

### Example

```
.PageInsertPages "Intro" , 4, 2, -1
```

The above example inserts 4 default pages in the Intro chapter before page 2.

---

{button ,AL('VENTURA\_Page\_Menu;;;','0',"Defaultoverview",)} Related Topics



## .FormatPubCondition (VENTURA)

**.FormatPubCondition** *.CondIndex=long, .Condition=string, .Enable=Boolean*

This command modifies a condition tag in the active document. Condition tags are used to identify text and frames specific to one version of the active document.

Syntax	Description
<b>.CondIndex</b>	Specifies a condition tag with an index number. The index numbers correspond to the condition tags listed in the Conditions list box in Corel VENTURA's Publications Properties dialog box (Click Format, Publication, Conditions). The first condition in the list is 1, the second condition in the list is 2, and so on.
<b>.Condition</b>	Specifies the new name of the condition tag. If omitted, the specified condition tag is not renamed.
<b>.Enable</b>	Specifies the status of the condition tag. To enable, set to TRUE (-1); otherwise, set to FALSE (0). If omitted, the status is unchanged.

### Note



This command cannot be recorded.



This command corresponds to Corel VENTURA's Conditions tab in the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Conditions.



You can count the number of conditions in a document with the [.PubConditionCount](#) function.



Use the [.FormatCondition](#) command to apply a condition to the active item in the active VENTURA document.

### Example

```
.FormatPubCondition 3, "ConNew", TRUE
```

The above example changes the name of the third condition to **ConNew** and sets its status to enabled.

---

{button ,AL("VENTURA\_FormatPubCondition\_Menu;vent\_pub;;;",0,"Defaultoverview"),} [Related Topics](#)




# .FormatPubConditionAdd (VENTURA)

**.FormatPubConditionAdd** *.Condition=string, .Enable=Boolean*

This command creates a condition tag in the active document. Condition tags are used to identify text and frames specific to one version of the active document.

Syntax	Description
<b>.Condition</b>	Specifies the name of the condition tag to add.
<b>.Enable</b>	Specifies whether to enable the new condition tag. To enable, set to TRUE (-1); otherwise, set to FALSE (0). If omitted, this parameter is set to FALSE.

**Note**

-  This command cannot be recorded.
-  This command corresponds to the Add button on the Conditions tab in the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Conditions.
-  Use the **.FormatCondition** command to apply a condition to the active item in the active VENTURA document.

**Example**

`.FormatPubCondition "Con1", TRUE`

The above example adds and enables the condition tag **Con1** in the active document.

---

`{button ,AL('VENTURA_FormatPubCondition_Menu;vent_pub;;;','0,"Defaultoverview",)} Related Topics`

## **.FormatPubConditionDelete (VENTURA)**

**.FormatPubConditionDelete** *.Condition=string*

This command deletes a condition tag from the active document. Any paragraphs or frames assigned the tag become unconditional.

Syntax	Description
<b>.Condition</b>	Specifies the name of the condition to delete.

### **Note**



This command cannot be recorded.



This command corresponds to the Delete button in the Conditions tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication.

### **Example**

`.FormatPubConditionDelete "Con1"`

The above example deletes the Con1 condition from the active document.

---

`{button ,AL("VENTURA_FormatPubConditionDelete_Menu;vent_pub;;;",0,"Defaultoverview"),}` [Related Topics](#)

## .FormatPubConditionGetAt (VENTURA)

**.FormatPubConditionGetAt** *.CondnlIndex=long*, *.Condition=string*, *.Enable=Boolean*

This function returns the name and status of a specified condition.

Syntax	Description
<b>.CondnlIndex</b>	Specifies a condition tag with an index number. The index numbers correspond to the condition tags listed in the Conditions list box in the Publications properties dialog box (Click Format, Publication, Conditions). The first condition in the list is 1, the second condition in the list is 2, and so on.
<b>.Condition</b>	Specifies a string variable that is assigned the name of the condition tag specified in <b>.CondnlIndex</b> .
<b>.Enable</b>	Specifies a string variable that is assigned the status of the condition tag specified in <b>.CondnlIndex</b> . If the condition tag is enabled, TRUE (-1) is assigned; otherwise, FALSE (0) is assigned.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.



You can count the number of conditions in a document with the **.PubConditionCount** function.

### Example

```
DIM Con_status AS BOOLEAN
.FormatPubConditionGetAt 2, ConTag$, Con_status
```

The above example assigns the condition tag name and its status to the Corel SCRIPT variables **ConTag** and **Con\_status**, respectively, for the second listed condition tag.

---

{button ,AL('VENTURA\_FormatPubConditionGetAt\_Menu;vent\_pub;;;',0,"Defaultoverview"),} Related Topics

## **.FormatPubConditionRename (VENTURA)**

**.FormatPubConditionRename** *.Condition=string, .NewName=string*

This command renames a condition tag in the active document.

Syntax	Description
<b>.Condition</b>	Specifies the condition tag to rename.
<b>.NewName</b>	Specifies the condition tag's new name.

### **Note**



This command cannot be recorded.



This command corresponds to the Rename button in the Conditions tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Conditions.

### **Example**

```
.FormatPubConditionRename "Con1", "Con4"
```

The above example renames Con1 to Con4.

---

{button ,AL("VENTURA\_FormatPubConditionRename\_Menu;vent\_pub;;;','0,"Defaultoverview"),} Related Topics

## **.FormatPubGeneral (VENTURA)**

**.FormatPubGeneral** .Layout=*long*, .DisableOverride=*Boolean*, .EmbedFont=*Boolean*, .PrintBlankPage=*long*

This command sets the general publication settings.

Syntax	Description
.Layout	Specifies the type of the layout. 0 full page 1 book 2 booklet 3 tent card 4 side fold card 5 top fold card
.DisableOverride	Set to TRUE (-1) to disable overrides. Set to FALSE (0) to enable overrides.
.EmbedFont	Set to TRUE (-1) to embed Bitstream TrueDoc fonts with the publication.
.PrintBlankPage	Specifies the option for printing a blank page between chapters, when necessary: 0 Ask user 1 Always insert 2 Never insert

### **Example**

```
.FormatPubGeneral 0, TRUE, FALSE, 1
```

This example sets the publication properties to full page layout; with overrides disabled; and automatic insertion of a blank page at the end of chapters that end on the same side of the page as the following chapter.

---

```
{button ,AL('VENTURA_PublicationName_Menu;vent_pub;;;',0,"Defaultoverview",)} Related Topics
```

## .FormatPubGeneralGet (VENTURA)

**.FormatPubGeneralGet** .Layout=*long* , .DisableOverride=*Boolean* , .EmbedFont=*Boolean* , .PrintBlankPage=*long*

This command gets the general settings for the current publication.

Syntax	Description
.Layout	Specifies the numeric variable which is assigned the publication layout: 0 full page 1 book 2 booklet 3 tent card 4 side fold card 5 top fold card
.DisableOverride	Specifies the numeric variable which is assigned the override setting: TRUE (-1) if disable overrides is set; otherwise FALSE (0).
.EmbedFont	Specifies the numeric variable which is assigned the Bitstream TrueDoc fonts setting: TRUE (-1) if embedded; otherwise FALSE (0).
.PrintBlankPage	Specifies the numeric variable which is assigned the option for printing a blank page between chapters, when necessary: 0 Ask user 1 Always insert 2 Never insert

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.

### Example

```
DIM DisableOverrides AS BOOLEAN
DIM EmbedFont AS BOOLEAN
.FormatPubGeneralGet Layout&, DisableOverrides, EmbedFont, PrintBlankPage&
MESSAGE Layout&
```

This example obtains the publication properties from the active publication.

---

{button ,AL('VENTURA\_PublicationName\_Menu;vent\_pub;;;','0,"Defaultoverview",)} Related Topics

## .FormatPubText (VENTURA)

**.FormatPubText** *.TextIndex=long*, *.TextName=string*, *.DefinedText=string*

This command modifies a Text Before/After text string in the active document. VENTURA automatically inserts the text string into a document when the text string's identifier is applied to a specific paragraph or character tag. Use this feature, for example, to insert the word "Warning" before the warning text. After defining the text string, assign the label identifying it to the paragraph or character tag which when applied to text inserts the text string.

Syntax	Description
<b>.TextIndex</b>	Specifies an identifier with an index number. The index numbers correspond to the order in which the text strings are created. The last created text string is 1, the second-last created text string is 2, and so on. The first created text string is equal to the total number of text strings. If a document variable is deleted, the index is recompiled.  You can count the number of text strings in a document with the <b>.PubTextCount</b> function.
<b>.TextName</b>	Specifies a new name for the text identifier. Omit this parameter if the identifier will not be modified.
<b>.DefinedText</b>	Specifies the text associated with the text identifier. Omit this parameter if the text will not be modified.

### Note



This command cannot be recorded.



This command corresponds to the Text Before/After tab in the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Text Before/After tab.

### Example

```
.FormatPubText 2, "NewName", "NewText"
```

The above example sets the second-last created text string identifier to NewName and its text to **NewText**.

---

{button ,AL('VENTURA\_FormatPubText\_Menu;vent\_pub;;;',0,"Defaultoverview",)} Related Topics



## .FormatPubTextAdd (VENTURA)

**.FormatPubTextAdd** .TextName=*string*, .DefinedText=*string*

This command adds a Text Before/After text string to the active document. VENTURA automatically inserts the text string into a document when the text string's identifier is applied to a specific paragraph or character tag. Use this feature, for example, to insert the word "Warning" before the warning text.

Syntax	Description
.TextName	Specifies the identifier of the Text Before/After text string.
.DefinedText	Specifies the text associated to the text identifier.

### Note



This command cannot be recorded.



This command corresponds to the Add button on the Text Before/After tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Text Before/After tab.

### Example

```
.FormatPubTextAdd "TBA", "This is Text Before/After text"
```

The above example adds a text string named TBA to the document. TBA is set to **This is Text Before/After text**.

---

{button ,AL("VENTURA\_FormatPubTextAdd\_Menu;vent\_pub;;;",0,"Defaultoverview"),} [Related Topics](#)

## **.FormatPubTextDelete (VENTURA)**

**.FormatPubTextDelete .TextName=*string***

This command deletes a Text Before/After text string in the active document.

Syntax	Description
<b>.TextName</b>	Specifies the identifier of the Text Before/After text string.

### **Note**



This command cannot be recorded.



This command corresponds to the Delete button on the Text Before/After tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Text Before/After tab.

### **Example**

**.FormatPubTextDelete "TBA1"**

The above example deletes the TBA1 text string from the active document.

---

**{button ,AL('VENTURA\_FormatPubTextDelete\_Menu;vent\_pub;;;',0,"Defaultoverview"),}** [Related Topics](#)

## .FormatPubTextGetAt (VENTURA)

**.FormatPubTextGetAt** *.TextIndex=long, .TextName=string, .DefinedText=string*

This function returns the identifier and the defined text of a specified Text Before/After text string in the active document.

Syntax	Description
<b>.TextIndex</b>	Specifies an identifier with an index number. The index numbers correspond to the order in which the text strings are created. The last created text string is 1, the second-last created text string is 2, and so on. The first created text string is equal to the total number of text strings. If a document variable is deleted, the index is recompiled.  You can count the number of text strings in a document with the <b>.PubTextCount</b> function.
<b>.TextName</b>	Specifies a string variable that is assigned the name of the text identifier. This identifier is specified in <b>.TextIndex</b> .
<b>.DefinedText</b>	Specifies a string variable that is assigned the Text Before/After text string of the text identifier. This identifier is specified in <b>.TextIndex</b> .

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatPubVariableGetAt 2, IdentName$, TBA$
```

The above example passes the text identifier and the Text Before/After text string to the variables **IdentName** and **TBA**, respectively, for the second-last created text string.

---

{button ,AL('VENTURA\_FormatPubTextGetAt\_Menu;vent\_pub;;;',0,"Defaultoverview"),} Related Topics

## **.FormatPubTextRename (VENTURA)**

**.FormatPubTextRename .TextName=string, .NewName=string**

This command renames a Text Before/After text string identifier in the active document.

Syntax	Description
<b>.TextName</b>	Specifies the identifier to rename.
<b>.NewName</b>	Specifies the identifier 's new name.

### **Note**



This command cannot be recorded.



This command corresponds to the Rename button on the Text Before/After tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Text Before/After tab.

### **Example**

```
.FormatPubTextRename "TBefore1", "TAfter1"
```

The above example renames the identifier TBefore1 to TAfter1.

---

{button ,AL("VENTURA\_FormatPubTextRename\_Menu;vent\_pub;;;','0,"Defaultoverview",,)} [Related Topics](#)

## .FormatPubVariable (VENTURA)

**.FormatPubVariable** *.VariableIndex=long*, *.VariableName=string*, *.Substitution=string*

This command modifies a variable in the active document. A variable is a placeholder for text that changes according to the variable's current definition. Variables are used to insert placeholders for items which are likely to change as you are preparing a document, such as dates, document version numbers, or names of customers and products.

Syntax	Description
<b>.VariableIndex</b>	Specifies a document variable with an index number. The index numbers correspond to the order in which the variables are created. The last created variable is 1, the second-last created variable is 2, and so on. The first created variable is equal to the total number of variables. If a document variable is deleted, the index is recompiled. You can count the number of variables in a document with the <b>.PubVariableCount</b> function.
<b>.VariableName</b>	Specifies a new name for the specified variable. Omit this parameter if the variable name is not to be modified.
<b>.Substitution</b>	Specifies the text to be substituted for the specified variable. Omit this parameter if the text is not to be modified.

### Note



This command cannot be recorded.



This command corresponds to the Variable Definition tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Variable Definition.

### Example

```
.FormatPubVariable 2, "NewName", "NewText"
```

The above example sets the second-last created document variable's name to **NewName** and its substitution text to **NewText**.

---

{button ,AL('VENTURA\_FormatPubVariable\_Menu;vent\_pub;;;','0,"Defaultoverview",')} [Related Topics](#)

## .FormatPubVariableAdd (VENTURA)

**.FormatPubVariableAdd .VariableName=string, .Substitution=string**

This command adds a variable to the active document. A variable is a placeholder for text that changes according to the variable's current definition. You use variables to insert placeholders for items which are likely to change as you are preparing a document, such as dates, document version numbers, or names of customers and products.

Syntax	Description
<b>.VariableName</b>	Specifies the variable name.
<b>.Substitution</b>	Specifies the text to be substituted for the specified variable.

### Note



This command cannot be recorded.



This command corresponds to the Add button in the Variable Definition tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Variable Definition.

### Example

```
.FormatPubVariableAdd "Var1", "This is sub text"
```

The above example adds the **Var1** variable to the document. **Var1** is set to **This is sub text**.

---

{button ,AL("VENTURA\_FormatPubVariableAdd\_Menu;vent\_pub;;;",0,"Defaultoverview"),} [Related Topics](#)

## .FormatPubVariableDelete (VENTURA)

**.FormatPubVariableDelete** *.VariableName=string*

This command deletes a variable definition from a document.

Syntax	Description
<b>.VariableName</b>	Specifies the name of the variable to delete.

### Note



This command cannot be recorded.



This command corresponds to the Delete button in the Variable Definition tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Variable Definition.

### Example

**.FormatPubVariableDelete** "Var1"

The above example deletes the **Var1** variable from the active document.

---

{button ,AL('VENTURA\_FormatPubVariableDelete\_Menu;vent\_pub;;;','0,"Defaultoverview",')} Related Topics

## **.FormatPubVariableGetAt (VENTURA)**

**.FormatPubVariableGetAt** *.VariableIndex=long*, *.VariableName=string*, *.Substitution=string*

This function returns the name and substitution text of a specified document variable.

Syntax	Description
<b>.VariableIndex</b>	Specifies a document variable with an index number. The index numbers correspond to the order in which the variables are created. The last created variable is 1, the second-last created variable is 2, and so on. The first created variable is equal to the number of variables. If a document variable is deleted, the index is recompiled. You can count the number of variables in a document with the <b>.PubVariableCount</b> function.
<b>.VariableName</b>	Specifies a string variable that is assigned the name of the document variable specified in <b>.VariableIndex</b> .
<b>.Substitution</b>	Specifies a string variable that is assigned the substitution text of the document variable specified in <b>.VariableIndex</b> .

### **Note**



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### **Example**

```
.FormatPubVariableGetAt 2, VarName$, VarSub$
```

The above example passes the variable name and substitution text to the variables **VarName** and **VarSub**, respectively, for the second-last created document variable.

---

{button ,AL("VENTURA\_FormatPubVariableGetAt\_Menu;vent\_pub;;;",0,"Defaultoverview",)} [Related Topics](#)



## **.FormatPubVariableRename (VENTURA)**

**.FormatPubVariableRename** *.VariableName=string, .NewName=string*

This command renames a variable in the active document.

Syntax	Description
<b>.VariableName</b>	Specifies the variable to rename.
<b>.NewName</b>	Specifies the variable's new name.

### **Note**



This command cannot be recorded.



This command corresponds to the Rename button in the Variable Definition tab of the Publication Properties dialog box in Corel VENTURA. Click Format, Publication, Variable Definition.

### **Example**

```
.FormatPubVariableRename "Var1", "Var100"
```

The above example renames Var1 to Var100.

---

{button ,AL("VENTURA\_FormatPubVariableRename\_Menu;vent\_pub;;;','0,"Defaultoverview"),} Related Topics

## **.FormatUpdatePublication (VENTURA)**

**.FormatUpdatePublication** *.AutoNumbering=Boolean, .TableOfContent=Boolean, .Index=Boolean, .EnvoyAcrobat=Boolean*

This command updates the table of contents, index, and auto-numbering of the current publication.

<b>Syntax</b>	<b>Description</b>
<code>.AutoNumbering</code>	Set to TRUE (-1) to update the auto-numbering. Otherwise, set to FALSE (0).
<code>.TableOfContent</code>	Set to TRUE (-1) to update the table of contents. Otherwise, set to FALSE (0).
<code>.Index</code>	Set to TRUE (-1) to update the index. Otherwise, set to FALSE (0).
<code>.EnvoyAcrobat</code>	Set to TRUE (-1) if you are going to publish to Adobe Acrobat (PDF). Otherwise, set to FALSE (0).

### **Examples**

```
.FormatUpdatePublication TRUE, FALSE, FALSE, FALSE
```

This example updates the auto-numbering in the current publication.

```
.FormatUpdatePublication FALSE, TRUE, TRUE, FALSE
```

This example updates the TOC and indices in the current publication.

---

`{button ,AL('VENTURA_FormatUpdatePublication_Menu;;;;',0,"Defaultoverview"),}` [Related Topics](#)


## .LoadStyleSheet (VENTURA)

**.LoadStyleSheet .FileName=string**

This command changes the style sheet used to format text in the active document. Tags in the new style sheet override those with the same name in the existing style sheet. Tags in the old style sheet that are not in the new one appear in uppercase letters in Corel VENTURA's Tags list. Formatting in the missing tags takes on the formatting of the Body Text tag in the new style sheet.

Syntax	Description
<b>.FileName</b>	Specifies the name and path of the style sheet to load. To load a style sheet from a pre-VENTURA 7 publication, specify a .STY file. To load a style sheet from another VENTURA 7 document, specify the document (usually a .VP file).

### Note

 This command corresponds to selecting a publication in the VENTURA Navigator in Publication Manager mode, right-clicking, and selecting Load Style Sheet.

### Example

```
.LoadStyleSheet "c:\Corel\ventura5\styles\menu.sty"
```

The above example loads the style sheet MENU.STY into the active document.

```
.LoadStyleSheet "c:\Corel\ventura7\docs\menu.vp"
```

The above example loads the style sheet from the MENU.VP document into the active document.

---

{button ,AL('vent\_pub;;;','0,"Defaultoverview",)} [Related Topics](#)


## .PubConditionCount (VENTURA)

**ReturnValue** = .PubConditionCount ( )

This function returns the number of condition tags in the active document.

Syntax	Description
<b>ReturnValue</b>	Specifies the numeric variable that is assigned the number of condition tags in the active VENTURA document.

### Note

 This command cannot be recorded.

### Example

```
ConCount& = .PubConditionCount ( )
```

The number of condition tags in the active document is assigned to **ConCount**.

---

{button ,AL('VENTURA\_PubConditionCount\_Menu;vent\_pub;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.PublicationModified (VENTURA)**

**ReturnValue** = .PublicationModified ()

This function returns a value indicating whether the active document has been modified without being saved.

Syntax	Description
<b>ReturnValue</b>	Specifies a numeric variable that is assigned a value indicating whether the active document has been modified without being saved. If the document is modified, TRUE (-1) is assigned; otherwise FALSE (0) is assigned.

### **Note**



This command cannot be recorded.

### **Example**

```
Mod = .PublicationModified ()
```

The above example assigns a Boolean value to the variable **Mod**.

---

{button ,AL(`vent\_pub;;;`,0,"Defaultoverview",)} [Related Topics](#)


## .PublicationName (VENTURA)

**ReturnString\$** = .PublicationName .DocIndex=*long*

This function returns the file name and path of the active document VENTURA document.

Syntax	Description
<b>ReturnString\$</b>	Specifies a string variable that is assigned the file name and path of the active document. If the active document has not ever been saved (a new document), the file title is returned (for example, "Publication 1").
.DocIndex	Specifies a numeric variable that is assigned the active documents index number. The index number corresponds to the numbers used for open documents on the VENTURA Window menu.

### Note

 This command cannot be recorded.

### Example

```
DocName$ = .PublicationName ()
```

The above example assigns the name and path of the active document to the **DocName** variable.

```
DocName$ = .PublicationName (DocIndex&)
```

This example assigns the name and path of the active document to the **DocName** variable, and it assigns the active document's index number to **DocIndex**.

---

{button ,AL('VENTURA\_PublicationName\_Menu;vent\_pub;;;','0,"Defaultoverview",,)} [Related Topics](#)


## .PubTextCount (VENTURA)

**Return Value** & = .PubTextCount ( )

This function returns the number of defined Text Before/After text strings in the active document.

Syntax	Description
<b>Return Value</b> &	Specifies the numeric variable that is assigned the number of defined Text Before/After text strings in the active document.

### Note

 This command cannot be recorded.

### Example

```
TextBACount& = .PubTextCount ( )
```

The number of defined Text Before/After text strings in the active document is assigned to **TextBACount**.

---

{button ,AL('VENTURA\_PubTextCount\_Menu;vent\_pub;;;','0,"Defaultoverview",,)} Related Topics

## .PubVariableCount (VENTURA)

**ReturnValue** = .PubVariableCount ( )

This function returns the number of variables in the active document.

Syntax	Description
<b>ReturnValue</b>	Specifies the numeric variable that is assigned the number of variables in the active VENTURA document.

### Note

 This command cannot be recorded.

### Example

```
VarCount& = .PubVariableCount ( )
```

The number of variables in the active document is assigned to **VarCount**.

---

{button ,AL('VENTURA\_PubVariableCount\_Menu;vent\_pub;;;',0,"Defaultoverview"),} [Related Topics](#)



## **.StyleSheetCopy (VENTURA)**

### **.StyleSheetCopy**

This command copies the active publication's style sheet to the clipboard. From the clipboard, the style sheet can be pasted into a VENTURA Library.

#### **Note**



This command corresponds to selecting a style sheet and right-clicking copy in the VENTURA Navigator in Publication Manager mode.

#### **Example**

```
.StyleSheetCopy  
.FileLibraryOpen "C:\Win95\Desktop\MyLib.vlb"  
.FileLibraryPaste
```

The above example copies the active style sheet to the clipboard, opens the MYLIB library, and pastes the copied style sheet into the library.

---

{button ,AL('VENTURA\_PublicationName\_Menu;vent\_pub;;;','0,"Defaultoverview",)} [Related Topics](#)

## .FormatSelectedText (VENTURA)

**.FormatSelectedText** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .Kerning=*long*, .ShiftUp=*long*, .Underline=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*, .Uppercase=*Boolean*, .Effect=*long*

This command changes the font, style (such as bold and italic), font size, color and other attributes for the selected text.

Syntax	Description																		
.FontName	Specifies the name of the font to use with the selected text. For a list of installed fonts, see the font box on the text or property toolbars.																		
.PointSize	Specifies the selected text's font size in points.																		
.Weight	<p>Specifies the font's weight setting (number of inked pixels per 1000 pixels). The weight setting ranges from 0 to 1000. Common values and their corresponding names include:</p> <table><tr><td>100</td><td>Thin</td></tr><tr><td>200</td><td>Extra Light, Ultra Light</td></tr><tr><td>300</td><td>Light</td></tr><tr><td>400</td><td>Normal, Regular</td></tr><tr><td>500</td><td>Medium</td></tr><tr><td>600</td><td>Semi Bold, Demi Bold</td></tr><tr><td>700</td><td>Bold</td></tr><tr><td>800</td><td>Extra Bold, Ultra Bold</td></tr><tr><td>900</td><td>Black, Heavy</td></tr></table> <p>This parameter is ignored if the font you're applying this setting to doesn't have the specified weight installed on your system. For example, Courier is normally available with only Normal (400) and Bold (700) weight settings. If you set the weight for a Courier font to a value other than 400 or 700, this parameter is ignored and the default setting is used.</p> <p>Some fonts may use different names than those shown above for the same weight settings. See a font dialog box for more information about the fonts you have installed on your system.</p>	100	Thin	200	Extra Light, Ultra Light	300	Light	400	Normal, Regular	500	Medium	600	Semi Bold, Demi Bold	700	Bold	800	Extra Bold, Ultra Bold	900	Black, Heavy
100	Thin																		
200	Extra Light, Ultra Light																		
300	Light																		
400	Normal, Regular																		
500	Medium																		
600	Semi Bold, Demi Bold																		
700	Bold																		
800	Extra Bold, Ultra Bold																		
900	Black, Heavy																		
.Italic	Specifies whether to apply italic formatting. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). You must have an italic version of the specified font installed on your system or this parameter is ignored.																		
.Kerning	<p>Specifies the amount of space to add or remove between the selected characters. A negative value removes space, placing the characters closer together. This setting is specified in hundredths of an em. An em is a unit of measure roughly equal to the width of the capital M at the current font size. Since the value is relative to the font size, you do not need to change the setting when the font size changes.</p> <p>Kerning affects the spacing between the selected characters and between the last of these characters and the one to the right of it.</p>																		
.ShiftUp	Specifies the distance to move the selected text off the baseline. A negative number moves the text down. This value is expressed in inches multiplied by 300, rounded to the nearest whole number. To express this value using points, multiply the number of points by 300, divide by 72, and round to the nearest whole number.																		
.Underline	<p>Specifies the selected text's font underline attributes:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Single</td></tr><tr><td>2</td><td>Double</td></tr><tr><td>3</td><td>Word underline</td></tr></table>	0	None	1	Single	2	Double	3	Word underline										
0	None																		
1	Single																		
2	Double																		
3	Word underline																		
.StrikeThru	Specifies the selected text's font strike-thru setting. Set to TRUE (-1) to enable strike-thru formatting. Set to FALSE (0) to disable strike-thru formatting.																		
.Overscore	Specifies the selected text's font overscore setting. Set to TRUE (-1) to enable overscore formatting. Set to FALSE (0) to disable italics formatting.																		
.Uppercase	Specifies whether to use uppercase characters with the selected text. Set to TRUE (-1) to enable uppercase characters; otherwise, set to FALSE (0).																		
.Effect	<p>Specifies the selected text's font effect attributes:</p> <table><tr><td>0</td><td>None</td></tr><tr><td>1</td><td>Subscript</td></tr><tr><td>2</td><td>Superscript</td></tr><tr><td>3</td><td>Alternate</td></tr></table>	0	None	1	Subscript	2	Superscript	3	Alternate										
0	None																		
1	Subscript																		
2	Superscript																		
3	Alternate																		

### Note



This command corresponds to the Text Properties dialog box in Corel VENTURA. Click Format, Text Properties.



Text must be selected for this command to work.



To use small caps, set the **Uppercase** parameter to TRUE (-1) and the **Effect** parameter to 3.

#### Example

```
.FormatSelectedText .FontName="Courier", .Underline=2
```

The above example sets the selected text to the "Courier" font with a double underline setting. This example could also be expressed as:

```
.FormatSelectedText "Courier", , , , , 2
```


---

{button ,AL("VENTURA\_FormatSelectedText\_Menu;vent\_selecttxt;;;',0,"Defaultoverview"),} [Related Topics](#)

# .FormatSelectedTextColor (VENTURA)

**.FormatSelectedTextColor** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This command sets the font color for selected text.

Syntax	Description
.ColorModel	<p>Specifies the color model to use for the selected text:</p> <ol style="list-style-type: none"><li>1 <a href="#">PANTONE Spot Colors</a></li><li>2 <a href="#">CMYK</a></li><li>3 <a href="#">CMYK255</a></li><li>4 <a href="#">CMY</a></li><li>5 <a href="#">RGB</a> (default if omitted)</li><li>6 <a href="#">HSB</a></li><li>7 <a href="#">HLS</a></li><li>8 <a href="#">Black and White</a></li><li>9 <a href="#">Grayscale</a></li><li>11 <a href="#">YIQ</a></li><li>12 <a href="#">L*a*b*</a></li><li>14 <a href="#">PANTONE Hexachrome</a></li></ol> <p>If this command is recorded, palettes are recorded as a color model. Click  for a list of equivalent color models.</p>
.Color1	<p>Specifies the first color component for the <b>.ColorModel</b> parameter. For example, Hue is the first color component for HSB. Click the green hotspot on the above color model for valid value ranges.</p>
.Color2	<p>Specifies the second color component for the <b>.ColorModel</b> parameter. For example, Green is the second color component for RGB. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.</p>
.Color3	<p>Specifies the third color component for the <b>.ColorModel</b> parameter. For example, Saturation is the third color component for HLS. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.</p>
.Color4	<p>Specifies the fourth color component for the <b>.ColorModel</b> parameter. For example, Black is the fourth color component for CMYK. Click the green hotspot on the above color model for valid value ranges. Omit this parameter if it is not available in the specified color model.</p>

## Note



This command corresponds to the Select Text Properties dialog box in Corel VENTURA. Click Format, Text.



Text must be selected for this command to work.

## Example

`.FormatSelectedText 5, 0, 30, 240`

The above example sets the selected text to a dark blue color using the RGB color model. The last parameter is omitted because RGB only uses 3 color components.

{button ,AL('VENTURA\_FormatSelectedTextColor\_Menu;vent\_selecttxt;;;','0,"Defaultoverview",)} [Related Topics](#)



# .FormatSelectedTextColorGet (VENTURA)

**.FormatSelectedTextColorGet** .ColorModel=*long*, .Color1=*long*, .Color2=*long*, .Color3=*long*, .Color4=*long*

This function returns the font color for selected text.

Syntax	Description
.ColorModel	Specifies the numeric variable that is assigned a return value corresponding to selected text's color model: 1 <a href="#">PANTONE Spot Colors</a> 2 <a href="#">CMYK</a> 3 <a href="#">CMYK255</a> 4 <a href="#">CMY</a> 5 <a href="#">RGB</a> (default if omitted) 6 <a href="#">HSB</a> 7 <a href="#">HLS</a> 8 <a href="#">Black and White</a> 9 <a href="#">Grayscale</a> 11 <a href="#">YIQ</a> 12 <a href="#">L*a*b*</a> 14 <a href="#">PANTONE Hexachrome</a>
.Color1	Specifies the numeric variable that is assigned the value for the first color component for <b>.ColorModel</b> . For example, Hue is the first color component for HSB. Click the green hot spot on the above color model for valid value ranges.
.Color2	Specifies the numeric variable that is assigned the value for the second color component for <b>.ColorModel</b> . For example, Green is the second color component for RGB. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color3	Specifies the numeric variable that is assigned the value for the third color component for <b>.ColorModel</b> . For example, Saturation is the third color component for HLS. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.
.Color4	Specifies the numeric variable that is assigned the value for the fourth color component for <b>.ColorModel</b> . For example, Black is the fourth color component for CMYK. Click the green hot spot on the above color model for valid value ranges. If this parameter is not available in the color model, 0 is returned.

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

## Example

```
.TypeText "Hello!"  
.TextCharLeft 1, -1  
.FormatSelectedTextColorGet cm&, c1&, c2&, c3&, c4&
```

The above example inserts the text "Hello!" and the selects the exclamation mark. The function the returns the color values for the exclamation mark to the variables **cm**, **c1**, **c2**, **c3**, and **c4**.

## .FormatSelectedTextGet (VENTURA)

**.FormatSelectedText** .FontName=*string*, .PointSize=*single*, .Weight=*long*, .Italic=*Boolean*, .Kerning=*long*, .ShiftUp=*long*, .Underline=*long*, .StrikeThru=*Boolean*, .Overscore=*Boolean*, .Uppercase=*Boolean*, .Effect=*long*

This function returns the font attributes for the selected text.

Syntax	Description
.FontName	Specifies the string variable that is assigned a return value corresponding to the selected text's font.
.PointSize	Specifies the numeric variable that is assigned a return value corresponding to the selected text's font point size.
.Weight	Specifies a numeric variable that is assigned the font's weight setting (number of inked pixels per 1000 pixels). The weight setting ranges from 0 to 1000. Common values and their corresponding names include: 100 Thin 200 Extra Light, Ultra Light 300 Light 400 Normal, Regular 500 Medium 600 Semi Bold, Demi Bold 700 Bold 800 Extra Bold, Ultra Bold 900 Black, Heavy
.Italic	Specifies a numeric variable that is assigned the font's italic setting: TRUE (-1) if enabled; FALSE (0) otherwise.
.Kerning	Specifies the numeric variable that is assigned a return value corresponding to the selected text's kerning setting. This setting is specified in hundredth's of an em. An em is a unit of measure roughly equal to the width of the capital M at the current font size.
.ShiftUp	Specifies the numeric variable that is assigned a return value corresponding to the selected text's baseline shift. This value is expressed in inches multiplied by 300, rounded to the nearest whole number.
.Underline	Specifies the numeric variable that is assigned a return value corresponding to the selected text's underline attributes: 0 None 1 Single 2 Double 3 Word underline
.StrikeThru	Specifies the numeric variable that is assigned a return value corresponding to the selected text's strike-thru setting: TRUE (-1) if enabled or FALSE (0) if disabled.
.Overscore	Specifies the numeric variable that is assigned a return value corresponding to the selected text's overscore setting: TRUE (-1) if enabled or FALSE (0) if disabled.
.Uppercase	Specifies the numeric variable that is assigned a return value corresponding to the selected text's uppercase setting: TRUE (-1) if enabled or FALSE (0) if disabled.
.Effect	Specifies the numeric variable that is assigned a return value corresponding to the selected text's effect attributes: 0 None 1 Subscript 2 Superscript 3 Small caps

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.TypeText "Hello!"  
.TextCharLeft 1, -1  
.FormatSelectedTextGet .FontName=fname$, .Underline=uline&
```

The above example inserts the text "Hello!" and the selects the exclamation mark. The function the returns the selected text's font and underline attributes to the variables **fname** and **uline**, respectively. This example can also be expressed as:

.FormatSelectedTextGet fname\$, , , , , uLine&

---

{button ,AL('VENTURA\_FormatSelectedTextGet\_Menu;vent\_selecttxt;;;',0,"Defaultoverview",)} Related Topics

## **.FormatSelectedTextReset (VENTURA)**

### **.FormatSelectedTextReset**

This command removes formatting from selected text. The selected text must have had formatting applied to it with VENTURA's text formatting toolbar, the Selected Text Properties dialog box in VENTURA (click Format, Text), or the [.FormatSelectedText](#) and [.FormatSelectedTextColor](#) scripting commands.

#### **Note**



This command corresponds to clicking the Normal button (



) in VENTURA.

#### **Example**

```
.FormatSelectedTextReset
```

---

```
{button ,AL('VENTURA_FormatSelectedTextGet_Menu;vent_selecttxt;;;','0,"Defaultoverview",)} Related Topics
```



## **.FormatSetFont (VENTURA)**

**.FormatSetFont** .FontName=*string*, .PointSize=*single*

This command changes the font, font size, or both, of selected text.

<b>Syntax</b>	<b>Description</b>
.FontName	Specifies the name of the font to use. For a list of installed fonts, see the font box on the text or property toolbars.
.PointSize	Specifies the font size in points.

### **Example**

```
.FormatSetFont "Courier", 16
```

The above example sets the font for selected text to Courier using 16 points.

---

{button ,AL("VENTURA\_FormatSelectedText\_Menu;vent\_selecttxt;;;",0,"Defaultoverview",,)} [Related Topics](#)

## **.TextCapitalize (VENTURA)**

### **.TextCapitalize**

This command converts the case of the selected text to initial capital letters (the first letter of each word is capitalized). This command corresponds to a right-click with text selected (Attribute, Capitalize).

#### **Note**



See [.TextLowercase](#) and [.TextUppercase](#) for other text conversion commands.



You can also use the [LCASE](#) or [UCASE](#) Corel SCRIPT commands to convert a string variable's case.

#### **Example**

```
.TypeText "ottawa"  
.TextWordLeft 1, -1  
.TextCapitalize
```

The above example inserts the text "ottawa" and then selects it. The last command changes the lowercase "o" to uppercase. The text then reads as "Ottawa".

---

{button ,AL('VENTURA\_TextCharLeft\_Menu;vent\_selecttxt;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.TextLowercase (VENTURA)**

### **.TextLowercase**

This command converts selected text to lowercase. This command corresponds to a right-click with text selected (Attribute, Lowercase).

#### **Note**



See [.TextCapitalize](#) and [.TextUppercase](#) for other text conversion commands.



You can also use the [LCASE](#) or [UCASE](#) Corel SCRIPT commands to convert a string variable's case.

#### **Example**

```
.TypeText "OTTAWA"  
.TextCharLeft 5, -1  
.TextLowercase
```

The above example inserts the text "OTTAWA" and then selects the last five characters. These characters are set to lowercase. The text then reads as "Ottawa".

---

{button ,AL('Ventrura\_TextLowercase\_Menu;vent\_selecttxt;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.TextUppercase (VENTURA)**

### **.TextUppercase**

This command converts selected text to uppercase. This command corresponds to a right-click with text selected (Attribute, Uppercase).

#### **Note**



See [.TextCapitalize](#) and [.TextLowercase](#) for other text conversion commands.



You can also use the [LCASE](#) or [UCASE](#) Corel SCRIPT commands to convert a string variable's case.

#### **Example**

```
.TypeText "ottawa"  
.TextWordLeft 1, -1  
.TextUppercase
```

The above example inserts the text "ottawa" and then selects it. These characters are set to uppercase. The text then reads as "OTTAWA".

---

{button ,AL('Ventrura\_TextUppercase\_Menu;vent\_selecttxt;;;','0,"Defaultoverview"),} [Related Topics](#)

## .IsCaretInTable (VENTURA)

ReturnValue = .IsCaretInTable ( )

This function returns a value indicating whether the insertion point is in a table.

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a return value corresponding to whether the insertion point is in a VENTURA table: TRUE (-1) in a table FALSE (0) not in a table

### Note

**I** This command cannot be recorded.

### Example

```
RV = .IsCaretInTable ( )  
IF RV = FALSE THEN .TableInsertTable 2, 3
```

In the above example, the first statement determines whether the insertion point is in a table (a return value is assigned to **RV**). The second statement is conditional and instructs VENTURA to insert a table if **RV** is equal to 0 (the insertion point is not in a table).

---

{button ,AL("VENTURA\_IsCaretInTable\_Menu;vent\_table;;;",0,"Defaultoverview",)} [Related Topics](#)


## **.TableAppendColumn (VENTURA)**


**.TableAppendColumn** .NumCols=*long*


This commands appends table columns at the right side of the active table.

Syntax	Description
.NumCols	Specifies the number of columns to append. Set to 1 if omitted.

### **Note**

 This command corresponds to changing the number of columns in the General tab of the Table Properties dialog box in Corel VENTURA. Click Table, Table Format.

 You can also add columns with the [.TableInsertColumn](#) command.

 You can append rows with the [.TableAppendRow](#) command.

### **Example**

`.TableAppendColumn 3`

The above example appends 3 columns at the right side of the table containing the insertion point.

---

`{button ,AL('VENTURA_TableCellBorders_Menu;vent_table;;;','0,"Defaultoverview",)}` [Related Topics](#)


# .TableAppendRow (VENTURA)


**.TableAppendRow** .NumRows=*long*


This commands appends table rows at the bottom of the active table.

Syntax	Description
.NumRows	Specifies the number of rows to append. Set to 1 if omitted.

**Note**

 This command corresponds to changing the number of rows in the General tab of the Table Properties dialog box in Corel VENTURA. Click Table, Table Format.

 You can also add rows with the [.TableInsertRow](#) command.

 You can append rows with the [.TableAppendColumn](#) command.

**Example**

`.TableAppendRow 4`

The above example appends 4 rows at the bottom of the table containing the insertion point.

---

`{button ,AL('VENTURA_TableCellBorders_Menu;vent_table;;;',0,"Defaultoverview"),}` [Related Topics](#)

## .TableAutoFill (VENTURA)

**.TableAutoFill** .Direction=*long*

This command automatically fills selected cells with a series of data by incrementing the value in the selected cells. You can use this command, for example, to automatically fill cells with the days of the week, or months of the year.

The following table shows examples of series created using this command.

Data in selected cells	Resulting series
1, 2	3, 4, 5, 6,...
1, 4	7, 10, 13, 16
May	June, July, August,...
01/01/96 (see Note)	02/01/96, 03/01/96, 04/01/96
text 1, text 2	text 3, text 4, text 5, text 6,...

Syntax	Description
.Direction	Specified the direction of the autofill: 0 Down (default if omitted) 1 Up 2 Left 3 Right

### Note



This command corresponds to the Autofill command on the Table menu in Corel VENTURA. Click Table, Autofill.



When entering dates in cells, you must use the Windows date format as specified through the Regional Settings in the Windows Control Panel.

### Example

```
.TableCreateTable 7, 4  
.TypeText "Sunday"  
.TableSelectColumn  
.TableAutoFill 0
```

The above example inserts a table 7 rows by 4 columns. **Sunday** is inserted in the top-left cell. The entire left column is then selected and an autofill is applied to it. The left column has all the days of the week included in it (Sunday, Monday, Tuesday,...).

---

{button ,AL("VENTURA\_TableAutoFill\_Menu;vent\_table;;;",0,"Defaultoverview",)} [Related Topics](#)



## .TableAutoFillSeries (VENTURA)

**.TableAutoFillSeries** .Direction=*long*, .Type=*long*, .DateUnit=*long*, .StepValue=*single*, .StopValue=*single*, .Trend=*Boolean*

This command, like the **.TableAutoFill** command automatically fills selected cells with a series of data by incrementing the value in the selected cells. It can also be used to increment values using trends, growth, and steps.

Syntax	Description
.Direction	Specifies the direction of the autofill: 0 Down (default if omitted) 1 Up 2 Left 3 Right
.Type	Specifies the autofill type: 0 AutoFill 1 Date 2 Growth 3 Linear  If this parameter is set to AutoFill, the parameters that follow are ignored.
.DateUnit	Specifies which part of the date is incremented for a date type autofill: 0 Day 1 Weekday 2 Month 3 Year  This parameter is only used if the <b>.Type</b> parameter is set to Date.
.StepValue	If <b>.Type</b> is set to Growth or Linear, specifies the amount by which to increase or decrease the series. A positive number increases a series; a negative number decreases a series. This can be a fractional value. If <b>.Trend</b> is not set to FALSE (0), this parameter is ignored.
.StopValue	If <b>.Type</b> is set to Growth or Linear, specifies the value at which the series ends. This can be a fractional value. If <b>.Trend</b> is not set to FALSE (0), this parameter is ignored.  If the selection is filled before the series reaches the stop value, the series stops at that point. This parameter is ignored with a date type autofill.
.Trend	If <b>.Type</b> is set to Growth or Linear, specifies the increment type: To use Trend increments, set to TRUE (-1); to use Step increments, set to FALSE (0) (default if omitted)

The following tables show how the Trend and Series options affect a linear or growth series using the values 1, 3 and 7 and 3 blank cells as an example. The first table shows the series resulting from using the Linear **.Type** option and setting **.StepValue** to 2.

Linear	Resulting series
.Trend is set to Step	1, 3, 5, 7, 9, 11
.Trend is set to Trend	0.66, 3.66, 6.66, 9.66, 12.66, 15.66

**Note** The StepValue is ignored when the creating a trend.

Using the same example with the Growth **.Type** option, produces the following series.

Growth	Resulting series
.Trend is set to Step	1, 2, 4, 8, 16, 32
.Trend is set to Trend	1.04, 2.75, 7.29, 19.31, 51.09, 135.18

### Note



This command corresponds to the Autofill Series command on the Table menu in Corel VENTURA. Click Table, Autofill, Series.

### Example

```
.TableCreateTable 6, 4
.TypeText "1"
.TextCharDown 1, 0
.TypeText "3"
.TextCharDown 1, 0
.TypeText "7"
```

```
.TableSelectColumn  
.TableAutoFillSeries 0, 2, , 2, , FALSE
```

The above example inserts a table 6 rows by 4 columns. Starting at the top of the first columns, the values 1, 3, and 7 are inserted into the column. The column is then selected. The **.TableAutoFillSeries** command is applied to column using a growth type and a step increment of 2. The resulting series is 1, 2, 4, 8, 16, 32.

---

{button ,AL('VENTURA\_TableCellBorders\_Menu;vent\_table;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.TableAutoFormat (VENTURA)**

**.TableAutoFormat .Style=long**

This command changes the format of the current table, based on the given preset style.

Syntax	Description
<b>.Style</b>	Specifies the table AutoFormat style (0-14). The table formats are listed in the AutoFormat dialog box (Click Table, AutoFormat)

### **Example**

**.TableAutoFormat 3**

This example changes the format of the current table based on the given preset style.

---

**{button ,AL('VENTURA\_TableCellBorders\_Menu;vent\_table;;;','0,"Defaultoverview",,)} Related Topics**

## **.TableAutoSum (VENTURA)**

### **.TableAutoSum**

This command inserts an automatic sum formula in a table cell.

#### **Example**

`.TableAutoSum`

---

`{button ,AL('VENTURA_TableCellBorders_Menu;vent_table;;;','0,"Defaultoverview",)} Related Topics`

## .TableCellBorders (VENTURA)

**.TableCellBorders** .Left=*string*, .Top=*string*, .Right=*string*, .Bottom=*string*, .AllEqual=*Boolean*

This command sets the borders of selected table cell(s).

Syntax	Description
.Left	Specifies the name of an existing border tag to apply to the left border.
.Top	Specifies the name of an existing border tag to apply to the top border.
.Right	Specifies the name of an existing border tag to apply to the right border.
.Bottom	Specifies the name of an existing border tag to apply to the bottom border.
.AllEqual	Specifies whether to make all borders equal. Set to TRUE (-1) to enable this option; otherwise FALSE (0) which is the default if omitted. If this option is enabled, one of the four other parameters must be specified. The borders are set to the first border parameter specified.

### Note



This command corresponds to the Cell Borders command in the Table menu in Corel VENTURA. Click Table, Cell Borders.



If more than one cell is selected, the selected cells are treated as one cell. The borders are only applied to the outside borders of the selection.

### Example

```
.TableCreateTable 10, 10  
.TableSelectTable 2, 3, 6, 4  
.TableCellBorders , "Z_THICK"
```

The above example inserts a table and then selects cells that intersect in the 5 rows (2 to 6) and 2 columns (3 to 4). The Z\_THICK border is applied to the top border of the cells at the top of the selection.

---

{button ,AL('VENTURA\_TableCellBorders\_Menu;vent\_table;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.TableColumnResize (VENTURA)**

**.TableColumnResize** *.Column=long .AmountX=long .ColumnStart=Boolean*

This command changes the width of the column by moving its left or right edge.

Syntax	Description
<b>.Column</b>	Specifies the column number to resize. The first column is 0, the second column is 1, etc.
<b>.AmountX</b>	Specifies the amount by which column edge has to be moved in tenths of a micorn. A positive value will move the column to the right. A negative value will move the column to the left.
<b>.ColumnStart</b>	Set to TRUE (-1) to move the column's starting (left) edge. Set to FALSE (0) to move the column's right edge. The default is FALSE.

### **Example**

```
.TableColumnResize 1, 254000, FALSE
```

This example changes the width of the first column by moving its right edge one inch.

---

{button ,AL('VENTURA\_TableCellBorders\_Menu;vent\_table;;;',0,"Defaultoverview"),} [Related Topics](#)

## .TableColumnWidth (VENTURA)

**.TableColumnWidth** **.Column=***long*, **.Fixed=***Boolean*, **.Width=***long*

This command sets the width attribute of a specified table column. The table must contain the insertion point.

Syntax	Description
<b>.Column</b>	Specifies column to resize. The first column is 0, the second column is 1, etc.
<b>.Fixed</b>	Specifies whether the specified column is made a particular size or proportional size. A proportional-width column uses whatever space is left over when fixed-width columns have been set.  If set to TRUE (-1), makes the specified column a fixed size. Set to FALSE (0) to make the specified column proportional in width. If omitted, set to FALSE.
<b>.Amount</b>	Specifies the width of the specified column, in tenths of a micron. If the <b>.Fixed</b> parameter is set to FALSE, this parameter is ignored.

### Note



This command corresponds to the General tab in the Table Properties dialog box in Corel VENTURA. Click Table, Format Table.



This command cannot be recorded.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.TableCreateTable 5, 6
.TableColumnWidth 3, -1, .5*M_INCH
```

The above example inserts a table and set the third column width to half an inch.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_TableColumnWidth\_Menu;vent\_table;;;",0,"Defaultoverview"),} Related Topics


# .TableColumnWidthGet (VENTURA)

.TableColumnWidthGet Column=*long*, .Fixed=*Boolean*, .Amount=*long*

This function returns the width of a specified column from the current table.

Syntax	Description
.Column	Specifies which column size is returned.
.Fixed	Specifies the numeric variable that is assigned whether the column width is fixed or proportional. If fixed, TRUE (-1); otherwise, FALSE (0) is returned.
.Amount	Specifies the numeric variable that is assigned the width of the specified column, in tenths of a micron.

## Note

 You must have the cursor planted in the table to use this command.

## Example

```
.TableColumnWidthGet 5, FP, ColWidth&
```

This example returns the width of the fifth column in the active table to the variable **ColWidth**. The **FP** variable is assigned a Boolean value indicating whether the fifth column is fixed or proportional.

---

{button ,AL('VENTURA\_TableCellBorders\_Menu;vent\_table;;;',0,"Defaultoverview",)} [Related Topics](#)




# .TableCountTables (VENTURA)

ReturnValue& =.TableCountTables ( )

This function counts the number of tables on the active page.

Syntax	Description
ReturnValue&	Specifies the numeric variable that is assigned the number of tables on the active page.

**Note**

 This command cannot be recorded.

**Example**

```
Tcount = .TableCountTables ( )
```

In the above example, the **Tcount** variable is assigned the number of tables on the active page.

---

{button ,AL('VENTURA\_TableCountTables\_Menu;vent\_table;;;','0,"Defaultoverview",)} [Related Topics](#)


## **.TableCreateTable (VENTURA)**


**.TableCreateTable .Rows=*long*, .Columns=*long***


This command inserts a table at the insertion point.

<b>Syntax</b>	<b>Description</b>
<b>.Rows</b>	Specifies the number of rows in the table.
<b>.Columns</b>	Specifies the number of columns in the table.

### **Note**

 This command corresponds to the Create Table command in the Table menu in Corel VENTURA. Click Table, Create Table.

 The table is inserted with default table properties. After a table has been inserted, the insertion point is found in the table's top-left hand corner cell.

 Tables cannot be inserted within tables.

### **Example**

`.TableCreateTable 2, 3`

The above example creates a table with two rows and three columns.

---

`{button ,AL('VENTURA_TableCreateTable_Menu;vent_table;;;',0,"Defaultoverview",)} Related Topics`


# .TableDeleteColumn (VENTURA)

`.TableDeleteColumn .NumColumns=long`

This command deletes table columns.

Syntax	Description
<code>.NumColumns</code>	<p>Specifies the number of columns to delete. If this parameter is set to 1, the column containing the insertion point is deleted. If this parameter is set to a value greater than one, the column containing the insertion point and the columns to the right of it are deleted (if exist).</p> <p>If this parameter is omitted and table cells are not selected, the column containing the insertion point is deleted. If this parameter is omitted and table cells are selected, the columns containing the selected cells are deleted.</p>

### Note

 This command corresponds to the Column command on the Delete flyout of the Table menu in Core! VENTURA. Click Table, Delete, Column.

 See [.TableDeleteTable](#) and [.TableDeleteRow](#) for other table deleting commands.

### Example

`.TableDeleteColumn`

The above example deletes the column containing the insertion point or the columns containing selected table cells.

`.TableDeleteColumn 5`

The above example deletes five columns from the table containing the insertion point.

---

`{button ,AL('VENTURA_TableDeleteColumn_Menu;vent_table;;;','0,"Defaultoverview",,)} Related Topics`


# .TableDeleteRow (VENTURA)

**.TableDeleteRow** .NumRows=*long*

This command deletes table rows.

Syntax	Description
.NumRows	Specifies the number of rows to delete. If this parameter is set to 1, the row containing the insertion point is deleted. If this parameter is set to a value greater than one, the row containing the insertion point and the rows below it are deleted (if exist).  If this parameter is omitted and table cells are not selected, the row containing the insertion point is deleted. If this parameter is omitted and table cells are selected, the rows containing the selected cells are deleted.

**Note**

 This command corresponds to the Row command on the Delete flyout of the Table menu in Corel VENTURA. Click Table, Delete, Row.

 See [.TableDeleteTable](#) and [.TableDeleteColumn](#) for other table deleting commands.

**Example**

`.TableDeleteRow`

The above example deletes the row containing the insertion point or the rows containing selected table cells.

`.TableDeleteRow 4`

The above example deletes four rows from the table containing the insertion point.

---

`{button ,AL('VENTURA_TableDeleteRow_Menu;vent_table;;;','0,"Defaultoverview",)} Related Topics`

## **.TableDeleteTable (VENTURA)**

### **.TableDeleteTable**

This command deletes the table containing the insertion point.

#### **Note**



This command corresponds to the Table command on the Delete flyout of the Table menu in Corel VENTURA. Click Table, Delete, Table.



See [.TableDeleteColumn](#) and [.TableDeleteRow](#) for other table deleting commands.

#### **Example**

```
.TableFirstTable  
.TableDeleteTable
```

The above example sends the insertion point to the first table on the current page and deletes that table.

---

**{button ,AL('VENTURA\_TableDeleteTable\_Menu;vent\_table;;;','0,"Defaultoverview",)} [Related Topics](#)**

## **.TableFirstTable (VENTURA)**

### **.TableFirstTable**

This command sends the insertion point to the first table on the active page. The insertion point is placed in the top left-hand corner table cell.

#### **Example**

```
.TableFirstTable  
.TableDeleteTable
```

The above example sends the insertion point to the first table on the current page and deletes that table.

---

{button ,AL('VENTURA\_TableFirstTable\_Menu;vent\_table;;;','0,"Defaultoverview",)} Related Topics


**.TableFormatTable (VENTURA)**


**.TableFormatTable** .Rows=*long*, .Columns=*long*, .Headers=*long*, .Footers=*long*, .AroundGrid=*string*, .HorizontalGrid=*string*, .VerticalGrid=*string*, .BreakAcrossPage=*Boolean*


This command sets the number of rows and columns, column widths, default borders and header and footer rows in the table containing the insertion point.

Syntax	Description
.Rows	Specifies the number of rows in the table. If the number of rows is reduced, the rows at the bottom of the table are removed.
.Columns	Specifies the number of columns in the table. If the number of columns is reduced, the columns on the right-hand side of the table are removed.
.Headers	Specifies the number of header rows in the table.
.Footers	Specifies the number of footer rows in the table.
.AroundGrid	Specifies the border style you want to apply to the perimeter of the table. For a list of border styles available in your publication, open the Table properties dialog box to the General tab (click Table, Format Table).
.HorizontalGrid	Specifies the border style you want to apply to the horizontal ruling lines. For a list of border styles available in your publication, open the Table properties dialog box to the General tab (click Table, Format Table).
.VerticalGrid	Specifies the border style you want to apply to the horizontal ruling lines. For a list of border styles available in your publication, open the Table properties dialog box to the General tab (click Table, Format Table).
.BreakAcrossPage	Specifies whether to break the table across pages if it is too large to fit on its current page. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0). If omitted, this parameter is set to FALSE.

**Note**

 This command corresponds to the General tab of the Table Properties dialog box in Corel VENTURA. Click Table, Table Format.

 To use this command, a table must either be selected or the insertion point must be in a table.

 If the number of columns is reduced, the overall width of the table stays the same and the remaining columns are resized.

**Example**

```
.TableCreateTable 4, 6
.TableFormatTable .Headers = 2, BreakAcrossPage = -1
```

The above example creates a table with four rows and six columns. The number of header rows in the table is set to 2 and the table is enabled to break across pages. This example could also be expressed as:

```
.TableCreateTable 4, 6
.TableFormatTable , , 2, , , , -1
```

## .TableFormatTableGet (VENTURA)

**.TableFormatTableGet** .Rows=*long* , .Columns=*long* , .Headers=*long* , .Footers=*long* , .AroundGrid=*string* , .HorizontalGrid=*string* , .VerticalGrid=*string* , .BreakAcrossPage=*Boolean*

This function returns the general settings of the current table.

Syntax	Description
.Rows	Specifies a numeric variable that is assigned the number of rows in the table.
.Columns	Specifies a numeric variable that is assigned the number of columns in the table.
.Headers	Specifies a numeric variable that is assigned the number of header rows in the table.
.Footers	Specifies a numeric variable that is assigned the number of footer rows in the table.
.AroundGrid	Specifies a string variable that is assigned the border style applied to the perimeter of the table.
.HorizontalGrid	Specifies a string variable that is assigned the border style applied to the horizontal ruling lines in the table.
.VerticalGrid	Specifies a string variable that is assigned the border style applied to the vertical ruling lines in the table.
.BreakAcrossPage	Specifies a numeric variable that is assigned a value indicating whether the table can break across pages if it is too large to fit on its current page. TRUE (-1) is returned if the table can break across pages; otherwise, FALSE (0).

### Note

**I** You must have the cursor planted in the table to use this command.

### Example

```
.TableFormatTableGet Rows&, Columns&, Headers&, Footers&, AroundGrid$, HorizontalGrid$,  
VerticalGrid$, BreakAcrossPage
```

```
MESSAGE Rows& & ", " & Columns& & CHR(13) & Headers& & ", " & Footers& & CHR(13) &  
AroundGrid$ & CHR(13) & HorizontalGrid$ & CHR(13) & VerticalGrid$ & CHR(13) &  
BreakAcrossPage
```

This example obtains the properties from the current table. The results are displayed in a message box.

---

{button ,AL('VENTURA\_TableCellBorders\_Menu;vent\_table;;;',0,"Defaultoverview"),} [Related Topics](#)




# .TableFunction (VENTURA)


.TableFunction .Formula=*string*

This command inserts a formula into the table cell containing the insertion point.

Syntax	Description
.Formula	Specifies, as a string, the formula to insert into the table cell.

**Note**

 This command corresponds to typing a formula in the Insert Formula box on Corel VENTURA's Table Property bar. It also corresponds to entering a formula using the Function Wizard dialog box. Click Table, Function.

 The first character in a formula string should be an equal sign (=).

**Example**

```
.TableGotoCell      'places insertion point in row 1, column 1
.TableFunction "=R1C4+R2C5"
```

The above example places the insertion point in the top left cell of the active table and inserts a formula into that cell. The formula adds the contents of cell **R1C4** and cell **R2C5** (where R indicates the row and C indicates the column).

```
.TableFunction "=ABS (R2C5) +SUM (R3C1:R11C1) "
```

The above example adds the absolute value of cell **R2C5** to the sum of nine cells in the first column.

---

{button ,AL('VENTURA\_TableFormatTable\_Menu;vent\_table;;;','0,"Defaultoverview",,)} Related Topics


# .TableFunctionGet (VENTURA)

ReturnString\$ = .TableFunctionGet ( )

This function returns the formula that is in the table cell where the insertion point resides.

Syntax	Description
ReturnString\$	Specifies the string variable that is assigned the formula. If the table cell does not contain a formula, an empty string is returned.

**Note**



The command cannot be recorded.

**Example**

```
.TableGotoCell      'places insertion point in row 1, column 1
FirstForm$ = .TableFunctionGet ( )
```

The above example places the insertion point in the top left cell of the active table. The formula in the cell is assigned to the **FirstForm** string variable.

## .TableGetCell (VENTURA)

**.TableGetCell** .Row=*long*, .Column=*long*

The function returns the position of the insertion point in a table.

Syntax	Description
.Row	Specifies the numeric variable that is assigned the table row the insertion point is in.
.Column	Specifies the numeric variable that is assigned the table column the insertion point is in.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.TableGetCell R&, C&
```

The row and column positions are assigned to the **R** and **C** variables, respectively.

---

{button ,AL('VENTURA\_TableGetCell\_Menu;vent\_table;;;',0,"Defaultoverview"),} Related Topics



**.TableGetSelection (VENTURA)**

**.TableGetSelection** .StartRow=*long*, .StartColumn=*long*, .EndRow=*long*, .EndColumn=*long*, .WholeTable=*Boolean*

This function returns the selected cell range in the table containing the insertion point.

Syntax	Description
.StartRow	Specifies the numeric variable that is assigned the top-most row of the selected table cells.
.StartColumn	Specifies the numeric variable that is assigned the left-most column of the selected table cells.
.EndRow	Specifies the numeric variable that is assigned the bottom-most row of the selected table cells.
.EndColumn	Specifies the numeric variable that is assigned the right-most column of the selected table cells.
.WholeTable	Specifies the numeric variable that is assigned a return value corresponding to whether the entire table containing the insertion point is selected: TRUE (-1)      entire table selected FALSE (0)      entire table not selected

**Note**

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

**Example**

`.TableGetSelection SR&, SC&, ER&, EC&, WT&`

In the above example, the table selection properties are assigned to five numeric variables.

`.TableGetSelection , , , , WT&`

In the above example, the **WT** variable is assigned a value corresponding to whether the entire table containing the insertion point is selected.

## **.TableGotoCell (VENTURA)**

**.TableGotoCell** .Row=*long*, .Column=*long*

This command places the insertion point in a specified table cell. The table must contain the insertion point before this command can be issued.

Syntax	Description
.Row	Specifies the row number of the cell to place the insertion point in. If omitted, the row number is set to 1.
.Column	Specifies the column number of the cell to place the insertion point in. If omitted, the column number is set to 1.

### **Example**

```
.TableGotoCell 2, 3 'places insertion point in row 2, column 3
.TableGotoCell , 3 'places insertion point in row 1, column 3
.TableGotoCell 2 'places insertion point in row 2, column 1
.TableGotoCell 'places insertion point in row 1, column 1
```

---

{button ,AL("VENTURA\_TableGotoCell\_Menu;vent\_table;;;','0,"Defaultoverview"),} [Related Topics](#)

## **.TableInsertColumn (VENTURA)**

**.TableInsertColumn** .NumCols=*long*, .Before=*Boolean*

This command inserts columns into a table containing the insertion point.

Syntax	Description
.NumCols	Specifies the number of columns to add. If omitted, the parameter is set to 1.
.Before	Specifies whether to insert the column(s) to the left or right of the insertion point. Set to TRUE (-1) to insert to the left of the insertion point. Set to FALSE (0) to insert to right of the insertion point. If omitted, the parameter is set to FALSE.

### **Note**



This command corresponds to the Insert Column command in the Table menu in Corel VENTURA. Click Table, Insert Column.



You can insert a row with the [.TableInsertRow](#) command.

### **Example**

```
.TableInsertColumn 2, -1
```

The above example inserts two columns to the left of the column containing the insertion point.

---

`{button ,AL("VENTURA_TableInsertColumn_Menu;vent_table;;;','0,"Defaultoverview"),}` [Related Topics](#)

## .TableInsertRow (VENTURA)

**.TableInsertRow** .NumRows=*long*, .Before=*Boolean*

This command inserts rows into a table containing the insertion point.

Syntax	Description
.NumCols	Specifies the number of rows to add. If omitted, the parameter is set to 1.
.Before	Specifies whether to insert the row(s) above or below the insertion point. Set to TRUE (-1) to insert rows above insertion point. Set to FALSE (0) to insert to rows below the insertion point. If omitted, the parameter is set to FALSE.

### Note



This command corresponds to the Insert Row command in the Table menu in Corel VENTURA. Click Table, Insert Row.



You can insert a column with the [.TableInsertColumn](#) command.

### Example

```
.TableInsertRow 4, 0
```

The above example inserts four rows below the row containing the insertion point.

---

{button ,AL('VENTURA\_TableInsertRow\_Menu;vent\_table;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.TableMergeCells (VENTURA)**

### **.TableMergeCells**

This command combines the selected table cells into one. The contents of the upper-left cell in the selection remains in the resulting larger cell. Cells can be divided with the [.TableSplitCells](#) command.

#### **Note**



This command corresponds to the Merge Cells command in the Table menu in Corel VENTURA. Click Table, Merge Cells.



You must have more than one cell selected before using this command.

#### **Example**

```
.TableCreateTable 2, 2  
.TableSelectRow  
.TableMergeCells
```

The above example inserts a table 2 rows by 2 columns. The first row is selected and the two cells are merged into one.

---

{button ,AL('VENTURA\_TableMergeCells\_Menu;vent\_table;;;','0,"Defaultoverview",')} [Related Topics](#)



## **.TableNextTable (VENTURA)**

### **.TableNextTable**

This command sends the insertion point to the next table on the active page. The insertion point is placed in the top-left hand corner table cell.

#### **Example**

```
.TableFirstTable  
.TableNextCell
```

The above example sends the insertion point to the first table on the active page and then to the next table on the active page.

---

{button ,AL("VENTURA\_TableNextTable\_Menu;vent\_table;;;','0,"Defaultoverview"),} Related Topics

## **.TableNormalRule (VENTURA)**

### **.TableNormalRule**

This command restores the rules (borders) applied to the selected table or cell when the table was first created.

#### **Note**



This command corresponds to the Normal Rules command on the right mouse button menu. Select a cell and click the right mouse button, click Normal Rules.

#### **Example**

`.TableNormalRule`

---

```
{button ,AL('VENTURA_TableNormalRule_Menu;vent_table;;;',0,"Defaultoverview"),} Related Topics
```

# .TablePosition (VENTURA)

**.TablePosition .CustomWidth=*long*, .Indent=*long*, .Alignment=*long***

This command sets a table to a width different from the width of column or frame containing the table. The table must contain the insertion point when this command is executed..

Syntax	Description
<b>.CustomWidth</b>	Specifies the overall width of the table, in tenths of a micron.
<b>.Indent</b>	Specifies the amount of space you want between the left edge of the column or frame and the table. This setting is ignored unless <b>.Alignment</b> is set Left.
<b>.Alignment</b>	Specifies the position of the table within the column or frame: 0 Left 1 Right 2 Center

**Note**

**I** This command corresponds to the Position tab on the Table Properties dialog box in Corel VENTURA. Click Table, Format Table, Position.

**I** You can set other table positioning attributes with the **.TableSpacing** command.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.TablePosition 5*M_INCH, 1*M_INCH, 0
```

The above example sets the table width to 5 inches. The table is left-aligned and is indented by 1 inch.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

# .TablePositionGet (VENTURA)

**.TablePositionGet** .CustomWidth=*long* , .Indent=*long* , .Alignment=*long*

This function returns the table width, alignment and indent settings of the selected table.

Syntax	Description
.CustomWidth	Specifies the numeric variable that is assigned the overall width of the table, in tenths of a micron.
.Indent	Specifies the numeric variable that is assigned the amount of space between the left edge of the column or frame and the table, in tenths of a micron.
.Alignment	Specifies the numeric variable that is assigned the position of the table within the column or frame: 0     Left 1     Right 2     Center

## Note

 You must have the cursor planted in the table to use this command.

## Example

```
.TablePositionGet CustomWidth&, Indent&, Alignment&  
MESSAGE TOINCHES (CustomWidth&) & CHR(13) & TOINCHES (Indent&) & CHR(13) & Alignment&
```

This example obtains the position properties from the current table. The units are converted from Corel VENTURA units (tenths of a micron) and displayed in a message box.

## **.TableRecalculate (VENTURA)**

### **.TableRecalculate**

This command updates the formulas in the table containing the insertion point.

#### **Note**



This command corresponds to the Recalculate command on the Table menu in Corel VENTURA. Click Table, Recalculate.

#### **Example**

```
.TableFirstTable  
.TableRecalculate
```

The above example sends the insertion point to the first table on the current page and recalculates all the formulas in the table.

---

`{button ,AL('VENTURA_TableRecalculate_Menu;vent_table;;;','0,"Defaultoverview",)} Related Topics`


## **.TableRemoveColumn (VENTURA)**


**.TableRemoveColumn** .NumCols=*long*


This commands removes table columns from the right side of the active table.

Syntax	Description
.NumCols	Specifies the number of columns to remove. Set to 1 if omitted.

### **Note**

 This command corresponds to changing the number of columns in the General tab of the Table Properties dialog box in Corel VENTURA. Click Table, Table Format.

 You can also remove columns with the [.TableDeleteColumn](#) command.

 You can remove rows with the [.TableRemoveRow](#) command.

### **Example**

`.TableRemoveColumn 3`

The above example removes 3 columns from the right side of the table containing the insertion point.

---

`{button ,AL('VENTURA_TableRecalculate_Menu;vent_table;;;','0,"Defaultoverview",)} Related Topics`


**.TableRemoveRow (VENTURA)**


**.TableRemoveRow** .NumRows=*long*


This commands removes table rows from the bottom of the active table.

Syntax	Description
.NumRows	Specifies the number of rows to remove. Set to 1 if omitted.

**Note**

 This command corresponds to changing the number of rows in the General tab of the Table Properties dialog box in Corel VENTURA. Click Table, Table Format.

 You can also remove rows with the [.TableDeleteRow](#) command.

 You can remove columns with the [.TableRemoveColumn](#) command.

**Example**

`.TableRemoveRow 4`

The above example removes 4 rows from the bottom of the table containing the insertion point.


## **.TableSelectCell (VENTURA)**

**.TableSelectCell** .SelectType=*long*, .Extend=*Boolean*

This command selects a cell in a table. The insertion point must be in a table to use this command.

Syntax	Description
.SelectType	Specifies the cell to select: 0     Cell containing insertion point (default if omitted) 1     Cell to right of the cell containing the insertion point 2     Cell to left of the cell containing insertion point
.Extend	This parameter only applies if table cells are already selected in the table. Set to TRUE (-1) to extend the selection of the already selected cells. Set to FALSE (0) to deselect the already selected cells.

### **Note**

 This command corresponds to the Cell command on the Select flyout of the right mouse button menu. Select a cell and click the right mouse button. Click Select, Cell.

### **Example**

```
.TableCreateTable 7, 4  
.TableSelectCell 1
```

The above example inserts a table 7 rows by 4 columns and selects the cell to the right of the table's top left cell.

---

{button ,AL("VENTURA\_TableSelectCell\_Menu;vent\_table;;;','0,"Defaultoverview"),} [Related Topics](#)




# .TableSelectColumn (VENTURA)

**.TableSelectColumn** .SelectType=*long*, .Extend=*Boolean*

This command selects a table column. The insertion point must be in a table to use this command.

Syntax	Description
.SelectType	Specifies the column to select: 0 Column containing insertion point (default if omitted) 1 Column to right of the column containing the insertion point 2 Column to left of the column containing insertion point
.Extend	This parameter only applies if table cells are already selected in the table. Set to TRUE (-1) to extend the selection of the already selected cells. Set to FALSE (0) to deselect the already selected cells.

### Note

 This command corresponds to the Column command on the Select flyout of the right mouse button menu. Select a cell and click the right mouse button. Click Select, Column.

 See the [.TableSelectRow](#) and [.TableSelectTable](#) commands for more information about selecting tables.

### Example

```
.TableCreateTable 7, 4
.TypeText "Sunday"
.TableSelectColumn
.TableAutoFill 0
```

The above example inserts a table 7 rows by 4 columns. "Sunday" is inserted in the top-left cell. The entire left column is then selected and an autofill is applied to it. The left column has all the days of the week included in it (Sunday, Monday, Tuesday,...).




# .TableSelectRow (VENTURA)

**.TableSelectRow** .SelectType=*long*, .Extend=*Boolean*

This command selects a table row. The insertion point must be in a table to use this command.

Syntax	Description
.SelectType	Specifies the row to select: 0     Row containing insertion point (default if omitted) 1     Row below the row containing insertion point 2     Row above the row containing insertion point
.Extend	This parameter only applies if table cells are already selected in the table. Set to TRUE (-1) to extend the selection of the already selected cells. Set to FALSE (0) to deselect the already-selected cells.

### Note

-  This command corresponds to the Row command on the Select flyout of the right mouse button menu. Select a cell and click the right mouse button. Click Select, Row.
-  See the [.TableSelectColumn](#) and [.TableSelectTable](#) commands for more information about selecting tables.
-  The insertion point must be in a table to use this command, or else an error occurs.

### Example

```
.TableCreateTable 4, 7
.TypeText "Sunday"
.TableSelectRow
.TableAutoFill 0
```

The above example inserts a table 4 rows by 7 columns. "Sunday" is inserted in the top left cell. The entire top row is then selected and an autofill is applied to it. The top row has all the days of the week included in it (Sunday, Monday, Tuesday,...).

## .TableSelectTable (VENTURA)

**.TableSelectTable** .StartRow=*long*, .StartColumn=*long*, .EndRow=*long*, .EndColumn=*long*, .WholeTable=*Boolean*

This command selects the entire table containing the insertion point. You can also use this command to select a range of columns and rows.

Syntax	Description
.StartRow	Specifies the first row to select in the table. If this parameter is omitted and the <b>.EndRow</b> parameter is used, <b>.StartRow</b> is set to 1.
.StartColumn	Specifies the first column to select in the table. If this parameter is omitted and the <b>.EndColumn</b> parameter is used, <b>.StartColumn</b> is set to 1.
.EndRow	Specifies the last row to select in the table. If this parameter is omitted and the <b>.StartRow</b> parameter is used, <b>.EndRow</b> is set to the row number of the last row in the table.
.EndColumn	Specifies the last column to select in the table. If this parameter is omitted and the <b>.StartColumn</b> parameter is used, <b>.EndColumn</b> is set to the column number of the last column in the table.
.WholeTable	Specifies whether to select the entire table containing the insertion point. Set to TRUE (-1) to select the entire table. Set to FALSE (0) or omit to not select the entire table. If this parameter is set to TRUE (-1), the first 4 parameters are ignored.

### Note

**I** This command corresponds to the Table command on the Select flyout of the right mouse button menu. Select a cell and click the right mouse button. Click Select, Table.

**I** If you use at least one column parameter and row parameter (with **.WholeTable** disabled), only the row cells and column cells that intersect are selected.

**I** See the [.TableSelectColumn](#) and [.TableSelectRow](#) commands for more information about selecting tables.

### Example

```
.TableCreateTable 10, 10  
.TableSelectTable , , , , -1
```

The above example inserts a table. The second line instructs VENTURA to select the entire table.

```
.TableCreateTable 10, 10  
.TableSelectTable 2, , 6
```

The above example inserts a table and then selects 5 rows (2 to 6).

```
.TableCreateTable 10, 10  
.TableSelectTable 2, 3, 6, 4
```

The above example inserts a table and then selects cells that intersect in the 5 rows (2 to 6) and 2 columns (3 to 4).

---

{button ,AL('VENTURA\_TableSelectTable\_Menu;vent\_table;;;','0,"Defaultoverview",)} [Related Topics](#)

## .TableSkew (VENTURA)

**.TableSkew** .SkewAngle=*long*, .RowHeight=*long*, .LeftColAngle=*long*, .SkewTextInCol=*Boolean*, .SkewFirstCell=*Boolean*, .ClipAtPage=*Boolean*

This command slants the top row, the left column, or both, for the table containing the insertion point.

Syntax	Description
<b>.SkewAngle</b>	Specifies the angle to slant the top row of cells in tenths of a degree (10 degrees equals 100 tenths of a degree).
<b>.RowHeight</b>	Specifies the maximum height of the top row of cells, in tenths of a micron.
<b>.LeftColAngle</b>	Specifies the angle to slant the left column cells in tenths of a degree (10 degrees equals 100 tenths of a degree).
<b>.SkewTextInCol</b>	Specifies whether to skew text in the left column. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). FALSE If omitted.
<b>.SkewFirstCell</b>	Specifies whether to skew the first cell of the table. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). FALSE If omitted.
<b>.ClipAtPage</b>	Specifies whether to clip on the page boundary. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0). FALSE If omitted.

### Note

**I** This command corresponds to the Skew tab on the Table Properties dialog box in Corel VENTURA. Click Table, Format Table, Skew.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.TableCreateTable 5, 6
.TableSkew 150, .5*M_INCH, 120, -1, -1, 0
```

The above example inserts a table and skews the top row 15 degrees and the left column 12 degrees. The top row height is set to half an inch, and the skew text in column and skip skew in first cell options are enabled.

The LENGTHCONVERT function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL("VENTURA\_TableSkew\_Menu;vent\_table;;;','0,"Defaultoverview"),} Related Topics

## .TableSkewGet (VENTURA)

**.TableSkewGet** .SkewAngle=*long* , .RowHeight=*long* , .LeftColAngle=*long* , .SkewTextInCol=*Boolean* , .SkewFirstCell=*Boolean* , .ClipAtPage=*Boolean*

This function returns the skew settings of the table.

Syntax	Description
.SkewAngle	Specifies the numeric variable that is assigned the slant angle of the top row of cells in tenths of a degree (10 degrees equals 100 tenths of a degree).
.RowHeight	Specifies the numeric variable that is assigned maximum height of the top row of cells, in tenths of a micron.
.LeftColAngle	Specifies the numeric variable that is assigned the slant angle of the left column cells in tenths of a degree (10 degrees equals 100 tenths of a degree).
.SkewTextInCol	Specifies the numeric variable that is assigned a value indicating whether the text in the left column is skewed. If skewed, TRUE (-1); otherwise, FALSE (0).
.SkewFirstCell	Specifies the numeric variable that is assigned a value indicating whether the first cell of the table is skewed. If skewed, TRUE (-1); otherwise, FALSE (0).
.ClipAtPage	Specifies the numeric variable that is assigned indicating whether the clip on the page boundary setting is enabled. If enabled, TRUE (-1); otherwise, FALSE (0).

### Note

**I** You must have the cursor planted in the table to use this command.

### Example

```
DIM SkewTextInCol AS BOOLEAN
DIM SkewFirstCell AS BOOLEAN
DIM ClipAtPage AS BOOLEAN
.TableSkewGet SkewAngle&, RowHeight&, LeftColAngle&, SkewTextInCol, SkewFirstCell, ClipAtPage

MESSAGE SkewAngle&/10 & CHR(13) & TOPOINTS(RowHeight&) & CHR(13) & LeftColAngle&/10 & CHR(13)
& SkewTextInCol & CHR(13) & SkewFirstCell & CHR(13) & ClipAtPage
```

This example obtains the skew properties from the current table. The units are converted from tenths of a micron and tenths of a degree(for angles) and displayed in a message box

---

{button ,AL('VENTURA\_TableCellBorders\_Menu;vent\_table;;;',0,"Defaultoverview",)} [Related Topics](#)


# .TableSkewTopRow (VENTURA)


**.TableSkewTopRow** *.Cell=long, .SkewText=Boolean*

This command slants text in a table's top row cells to the same angle as the cell. The insertion point must reside in the table for this command to work properly.

Syntax	Description
<b>.Cell</b>	Specifies the cell in the top row. The first cell is 0, the second cell is 1, and so on.
<b>.SkewText</b>	Specifies whether to skew the text. Set to TRUE (-1) to skew the text; otherwise set to FALSE (0), which is the default if omitted.

**Note**

 This command corresponds to the "Skew Text In Top Row" section of the Skew tab in the Table Properties dialog box in Corel VENTURA. Click Table, Format Table, Skew.

 To skew a table's top-row cell, see the [.TableSkew](#) command.

**Example**

```
.TableCreateTable 5, 6
.TableSkew 150, .5*M_INCH
.TableSkewTopRow 2, -1
```

The above example inserts a table and skews the top-row 15 degrees and sets the top row height to half an inch. The third cell in the top row then has its contents skewed to the same angle as the cell.

The [LENGHTCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

## .TableSort (VENTURA)

**.TableSort** .SortKey1=*long*, .KeyType1=*long*, .Descending1=*Boolean*, .CaseSensitive1=*Boolean*, .SortKey2=*long*, .KeyType2=*long*, .Descending2=*Boolean*, .CaseSensitive2=*Boolean*, .SortKey3=*long*, .KeyType3=*long*, .Descending3=*Boolean*, .CaseSensitive3=*Boolean*, .ColumnOnly=*Boolean*

This command sorts information in the selected column(s) alphabetically, numerically, or by date.

Syntax	Description
.SortKey1	Specifies which column you want to use as the first sort key. If omitted, the first column is used.
.KeyType1	Specifies the first sort key type: 0     Text (default if omitted) 1     Number 2     Date
.Descending1	Specifies the first key sort order: set to TRUE (-1) for descending order; set to FALSE (0) for ascending order (FALSE is the default if omitted).
.CaseSensitive1	Specifies whether to sort a capitalized letter before a lowercase letter for the first sort key: set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted. This option is only valid if the <b>.Descending1</b> parameter is set to Text.
.SortKey2	Specify a second column to use as the second sort key. The next three parameters are ignored if this parameter is omitted.
.KeyType2	Specifies the second sort key type: 0     Text (default if omitted) 1     Number 2     Date
.Descending2	Specifies the second key sort order: set to TRUE (-1) for descending order; set to FALSE (0) for ascending order (FALSE is the default if omitted).
.CaseSensitive2	Specifies whether to sort a capitalized letter before a lowercase letter for the second sort key: set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted. This option is only valid if the <b>.Descending2</b> parameter is set to Text.
.SortKey3	Specify a third column to use as the third sort key. The next three parameters are ignored if this parameter is omitted.
.KeyType3	Specifies the third sort key type: 0     Text (default if omitted) 1     Number 2     Date
.Descending3	Specifies the third key sort order: set to TRUE (-1) for descending order; set to FALSE (0) for ascending order (FALSE is the default if omitted).
.CaseSensitive3	Specifies whether to sort a capitalized letter before a lowercase letter for the third sort key: set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted. This option is only valid if the <b>.Descending3</b> parameter is set to Text.
.ColumnOnly	Specifies whether to sort only the selected column(s): set to TRUE (-1) to enable this option; otherwise set to FALSE (0), which is the default if omitted.

### Note



This command corresponds to the Sort dialog box in Corel VENTURA. Click Table, Sort.

### Example

```
.TableFirstTable
.TableSelectTable 1, 1, 4, 3
.TableSort 1, 0, FALSE, FALSE, 3, 1, TRUE, TRUE
```

The above example sends the insertion point to the first table on the current page and selects some columns. The table is then sorted: the first sort key is the first column using an ascending text sort with case-sensitivity disabled; the second sort key is the third column using a descending numeric sort.

---

{button ,AL('VENTURA\_TableSkewColumn\_Menu;vent\_table;;;','0,"Defaultoverview",)} [Related Topics](#)

## .TableSpacing (VENTURA)

**.TableSpacing** .AboveSpacing=*long*, .BelowSpacing=*long*, .InterRow=*long*, .InterCol=*long*, .VJTop=*long*, .VJBottom=*long*

This command sets the spacing, alignment and justification settings for the table containing the insertion point.

Syntax	Description
.AboveSpacing	Specifies the spacing above the selected table, in tenths of a micron.
.BelowSpacing	Specifies the spacing below the selected table, in tenths of a micron.
.InterRow	Specifies the space between rows in the selected table.
.InterCol	Specifies the space between columns in the selected table.
.VJTop	Specifies the maximum amount of space to be used at the top of the table when vertical justification is incurred.
.VJBottom	Specifies the maximum amount of space to be used at the bottom of the table when vertical justification is incurred.

### Note



You can set other table positioning attributes with the **.TablePosition** command.

### Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.TableCreateTable 5, 6
.TableSpacing .5*M_INCH, .5*M_INCH, .10*M_INCH, .10*M_INCH,
```

The above example inserts a table. The above and below table spacing is set to 0.5 inches and the inter-row and inter-column spacing is set to 0.1 inches.

The **LENGTHCONVERT** function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

---

{button ,AL('VENTURA\_TableSpacing\_Menu;vent\_table;;;','0,"Defaultoverview",)} [Related Topics](#)



## **.TableSpacingGet (VENTURA)**

**.TableSpacingGet** .AboveSpacing=*long* , .BelowSpacing=*long* , .InterRow=*long* , .InterCol=*long* , .VJTop=*long* , .VJBottom=*long*

This function returns the spacing settings of the current table.

Syntax	Description
.AboveSpacing	Specifies the numeric variable that is assigned the spacing above the selected table, in tenths of a micron.
.BelowSpacing	Specifies the numeric variable that is assigned the spacing below the selected table, in tenths of a micron.
.InterRow	Specifies the numeric variable that is assigned the space between rows in the selected table.
.InterCol	Specifies the numeric variable that is assigned the space between columns in the selected table.
.VJTop	Specifies the numeric variable that is assigned the maximum amount of space to be used at the top of the table when vertical justification is incurred.
.VJBottom	Specifies the numeric variable that is assigned the maximum amount of space to be used at the bottom of the table when vertical justification is incurred.

### **Note**

 You must have the cursor planted in the table to use this command.

### **Example**

```
.TableSpacingGet AboveSpacing&, BelowSpacing&, InterRow&, InterCol&, VJTop&, VJBottom&
```

```
MESSAGE TOPOINTS(AboveSpacing&) & CHR(13) & TOPOINTS(BelowSpacing&) & CHR(13) &  
TOPOINTS(InterRow&) & CHR(13) & TOPOINTS(InterCol&) & CHR(13) & TOPOINTS(VJTop&) & CHR(13) &  
TOPOINTS(VJBottom&)
```

This example obtains the spacing properties from the current table. The units are converted from tenths of a micron (VENTURA units) and displayed in a message box

---

{button ,AL('VENTURA\_TableCellBorders\_Menu;vent\_table;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.TableSplitCells (VENTURA)**

### **.TableSplitCells**

This command divides selected cells that have been merged.

#### **Note**



This command corresponds to the Split Cells command in the Table menu in Corel VENTURA. Click Table, Split Cells.

#### **Example**

.TableSplitCells

---

{button ,AL('VENTURA\_TableSplitCells\_Menu;vent\_table;;;','0,"Defaultoverview",')} Related Topics

## .CurrentTextFile (VENTURA)

ReturnString\$ =.CurrentTextFile ( )

This function returns the name of the text file that the insertion point resides in.

Syntax	Description
ReturnString\$	Specifies the string variable that is assigned the name of the current text file. If the text file is Export On Save enabled, the entire path and extension of the text file (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties) is assigned. If the text file is embedded, the file name which appears in the File List or the VENTURA Navigator (no extension is used) is assigned.  This parameter was modified in Corel VENTURA 8.

### Note



This command cannot be recorded.



The text file name this function returns corresponds to the name displayed in the VENTURA Navigator in Publication Manager mode.

### Example

```
Tname$ = .CurrentTextFile ( )
```

In the above example, the **TName** variable is assigned the name of the text file containing the insertion point.

---

{button ,AL('VENTURA\_TextCharRight\_Menu;vent\_text;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.DeleteCharacter (VENTURA)**

**.DeleteCharacter** .Before=*Boolean*

This command deletes a single character before or after the insertion point.

Syntax	Description
.Before	Specifies whether to delete the character before or after the insertion point. Set to TRUE (-1) to delete the character before. Set to FALSE (0) to delete the character after (default if omitted).

### **Note**



Spaces count as characters.



This command corresponds to pressing the BACKSPACE or DELETE in Core! VENTURA.

### **Example**

```
.TextCharDown 7  
.DeleteCharacter
```

The above example moves the insertion point down seven lines and deletes the character after the insertion point.

---

{button ,AL('VENTURA\_TextFileCount\_Menu;vent\_text;;;',0,"Defaultoverview"),} [Related Topics](#)



# .FormatAttachTextFile (VENTURA)

.FormatAttachTextFile .FileName=*string*

This command attaches a specified text file to the currently selected frame. The text file must already exist in the chapter before it can be attached.

Syntax	Description
.FileName	Specifies the name of the text file to attach. If the text file is Export On Save enabled, use the entire path and extension of the text file (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used).  This parameter was modified in Corel VENTURA 8.

Note

-  You can detach a text file from a frame using the [.FormatDetachTextFile](#) command.
-  You can attach a picture file using the [.FormatAttachPicture](#) command.

Example

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.FileImportText "C:\MYDOCS\REPORT.WP6", 12
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatAttachTextFile "REPORT"
```

The above example imports the Corel WordPerfect document REPORT.WP6 into the active chapter. The [.SelectObjectAt](#) command is used to select an existing frame, and the imported WordPerfect document is attached to it.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.



# .FormatDetachTextFile (VENTURA)

**.FormatDetachTextFile** .FileName=*string*

This command removes a specified text file from the currently selected frame and chapter.

Syntax	Description
.FileName	<p>Specifies the name of the text file to detach. If the text file is Export On Save enabled, use the entire path and extension of the text file (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used).</p> <p>This parameter was modified in Corel VENTURA 8.</p> <p>If not specified, the text file in the active frame is detached.</p>

**Note**

-  You can attach a text file to a frame using the [.FormatAttachTextFile](#) command.
-  You can detach a picture file using the [.FormatDetachPicture](#) command.

**Example**

```
M_INCH = LENGTHCONVERT (1 , 7 , 1)
.SelectObjectAt 2.5*M_INCH, 3*M_INCH
.FormatDetachTextFile "REPORT"
```

The above example selects an existing frame and removes the text file REPORT from it.

The [LENGTHCONVERT](#) function creates a variable (**M\_INCH**) that is equal to the number of tenths of a micron in an inch.

## .GetTextEntryMode (VENTURA)

ReturnValue =.GetTextEntryMode ( )

This function returns the current text entry mode (Override or Tag mode).

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a value indicating the text entry mode: TRUE (-1) for Override mode; FALSE (0) for Tag mode.

### Note



The [.SetTextEntryMode](#) command sets text entry mode.



The [.GetSelectionType](#) function can return the status of the mouse pointer: selection mode (Pick Tool) or text mode (I-beam)



This command cannot be recorded.

### Example

```
CurrentMode = .GetTextEntryMode ( )
```

The above example assigns the text entry mode to **CurrentMode** variable.

---

{button ,AL("VENTURA\_TextFileCount\_Menu;vent\_text;;;",0,"Defaultoverview",)} [Related Topics](#)

## **.InsertLineBreak (VENTURA)**

### **.InsertLineBreak**

This command inserts a line break in a paragraph. This command corresponds to pressing SHIFT+ENTER in VENTURA. A line break is sometimes called a soft return.

#### **Note**



The insertion point must be placed in text before you can use this command.



You can also insert a line break with the **.TypeText CHR(13)** command. See the **.TypeText** command for more information.

#### **Example**

`.InsertLineBreak`

---

`{button ,AL('VENTURA_InsertLineBreak_Menu;vent_text;;;','0,"Defaultoverview",)} Related Topics`



## **.InsertParagraph (VENTURA)**

### **.InsertParagraph**

This command inserts a paragraph mark in a paragraph. This command corresponds to pressing ENTER in VENTURA.

#### **Note**



This command corresponds to pressing ENTER in VENTURA.



The insertion point must be placed in text before you can use this command.



You can also insert a line break with the **.TypeText CHR(10)** command. See the **.TypeText** command for more information.

#### **Example**

**.InsertParagraph**

---

**{button ,AL("VENTURA\_InsertPara\_Menu;vent\_text;;;','0,"Defaultoverview"),}** [Related Topics](#)

## .SelectedText (VENTURA)

**ReturnString\$ = .SelectedText ( )**

This functions returns the currently selected text. Special characters and symbols are assigned in ANSI format. The resulting text is just as it would appear if the selected text was exported using the ANSI text filter.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the selected text.

### Note

**I** This command cannot be recorded.

### Example

```
.TextStartOfColumn  
.TextSelectWord  
FirstWord$ = .SelectedText ( )
```


The above example moves the insertion point to the beginning of the current text column and then selects the first word in the column. The first word in the column is assigned to the **FirstWord** string variable.

---

{button ,AL('VENTURA\_InsertPara\_Menu;vent\_text;;;','0,"Defaultoverview",)} Related Topics


## **.SetTextEntryMode (VENTURA)**

**.SetTextEntryMode .Override=Boolean**

This command sets the text entry mode in VENTURA: Tag mode or Override mode. This command corresponds to clicking the  button.

Syntax	Description
<b>.Override</b>	Specified the text entry mode. Set to TRUE (-1) for Override mode; set to FALSE (0) for Tag mode.

### **Note**

 The [.GetTextEntryMode](#) function returns the mode setting.

### **Example**

```
.SetTextEntryMode FALSE  
.SetTextEntryMode TRUE
```

In the above example the first line sets the text entry mode to Tag mode. The second line sets it to Override mode.

---

{button ,AL('VENTURA\_TextFileCount\_Menu;vent\_text;;;','0,"Defaultoverview",,)} [Related Topics](#)

## .TextFileCopy (VENTURA)

**.TextFileCopy** .FileName=*string*

Copies a specified text file in the active chapter to the clipboard.

Syntax	Description
<b>.FileName</b>	Specifies the name of the text file to export. If the text file is Export On Save enabled, use the entire path and extension of the text file (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used).  This parameter was modified in Corel VENTURA 8.

### Note



This command corresponds to selecting a text file in the VENTURA Navigator, right-clicking and selecting Copy.



The clipboard contents can be pasted into a VENTURA library, a frame, or another chapter.



Use the **.EditCopy** command to copy a selection.

### Example

```
.FileImportText "C:\MYDOCS\REPORT.WP6", 12  
.TextFileCopy "REPORT"
```

The above example imports the Corel WordPerfect REPORT.WP6 into the active chapter. The WordPerfect document is then copied to the clipboard.

---

{button ,AL('VENTURA\_TextFileCopy\_Menu;vent\_text;;;','0,"Defaultoverview",)} [Related Topics](#)

## .TextFileCount (VENTURA)

**ReturnValue& = .TextFileCount ( )**

This function returns the number of text files in the active chapter.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the number of text files in the active chapter.

### Note

**I** This command cannot be recorded.

### Example

```
TextFiles& = .TextFileCount ( )
```

The above example assigns the number of text files in the active chapter to the **TextFiles** variable.

---

{button ,AL('VENTURA\_TextFileCount\_Menu;vent\_text;;;','0,"Defaultoverview",,)} [Related Topics](#)

## .TextFileGetAt (VENTURA)

**ReturnString\$ = .TextFileGetAt .FileIndex=*long***

This function returns the name of a specified text file in the active Corel VENTURA chapter.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of the specified text file. If the text file is Export On Save enabled, the text file name and extension (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties) is assigned. If the text file is embedded, the file name which appears in the File List or the VENTURA Navigator (no extension is used) is assigned.  This parameter was modified in Corel VENTURA 8.
<b>.FileIndex</b>	Specifies the text file as an index number. The first text file corresponds to 1, the second text file corresponds to 2, and so on. The index numbers are based on the order in which the files are inserted into the chapter. If a text file is deleted from the chapter, the index numbers are re-ordered.

### Note



This function cannot be recorded.



This function returns the full path of the text file if it is linked (Export On Save option is enabled). You can enable this option using the [.FileRenameTextFile](#) command.

### Example

```
TestName$ = .TextFileGetAt (7)
```

In the above example, the name of index text file 7 is assigned to the variable **TestName**.

```
NumberOfTextFiles% = .TextFileCount ( )  
FOR i% = 1 TO NumberOfTextFiles%  
    TextFileName$(i%) = .TextFileGetAt(i%)  
NEXT i%
```

In this example, the number of text files in the active chapter are used in a FOR...NEXT LOOP to create a string array of all the text files in the active chapter.

---

{button ,AL('VENTURA\_TextFileGetAt\_Menu;vent\_text;;;',0,"Defaultoverview"),} [Related Topics](#)

## .TextFileIsUsed (VENTURA)

ReturnString = .TextFileIsUsed .FileName=*string*

This function returns a value indicating whether a specified text file in the active chapter has its contents inserted into a frame.

Syntax	Description
ReturnString	Specifies a numeric variable that is assigned a value indicating whether a specified text file has its contents inserted into a frame: TRUE (-1) if the file has its contents inserted; otherwise FALSE (0).
.FileName	Specifies the name of the text file to test. If the text file is Export On Save enabled, use the text file name and extension (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used). This parameter was modified in Corel VENTURA 8.

### Note



This command cannot be recorded.



You must have something selected to use this command.

### Example

```
TextFiles& = .TextFileCount( )
FOR i% = 1 TO TextFiles&
    TextName$ = .TextFileGetAt (i%)
    InUseStatus = .TextFileIsUsed (TextName$)

    IF InUseStatus = TRUE THEN MESSAGE "This text file is used"
    IF InUseStatus = FALSE THEN MESSAGE "This text file is not used"
NEXT i%
```

The above example tests whether the text files in the active chapter are used. The first line assigns the number of text files in the active chapter to the **TextFiles** variable. This variable is then used to set the maximum looping value. The **i%** variable is assigned to the **.TextFileGetAt** function as an index number. This function returns the name each text file in the active chapter. Each text file is then tested for use in the chapter.

---

{button ,AL('VENTURA\_TextFileCount\_Menu;vent\_text;;;',0,"Defaultoverview",)} [Related Topics](#)

## .TextFileParaCount (VENTURA)

**ReturnValue**& = .TextFileParaCount .FileName=*string*

Returns the number of paragraphs in a specified text file in the active chapter.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the number of paragraphs in the specified text file in the active chapter.
<b>.FileName</b>	Specifies the name of the text file to count in. If the text file is Export On Save enabled, use the text file name and extension (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used).  This parameter was modified in Corel VENTURA 8.

### Note



This command cannot be recorded.

### Example

```
TextPC = .TextFileParaCount "Intro.txt"
```

In the above example, the number of paragraphs in the INTRO.TXT file is assigned to the variable **TextPC**.

```
TextPC = .TextFileParaCount ( )
```

In the above example, the number of paragraphs in the active text file is assigned to the variable **TextPC**.

---

{button ,AL('VENTURA\_TextFileParaCount\_Menu;vent\_text;;;',0,"Defaultoverview"),} [Related Topics](#)




# .TextIsParaOverriden (VENTURA)

ReturnValue = .TextIsParaOverriden ( )

This function determines whether the current text's paragraph tag properties are overridden. The current text is the paragraph where the insertion point resides.

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a value indicating whether the paragraph tag properties are overridden: TRUE (-1) if overridden; otherwise FALSE (0).

Note



This command cannot be recorded.

Example

```
DIM TestOver AS BOOLEAN
TestOver = .TextIsParaOverriden ( )
```

In the above example, the **TestOver** variable is assigned a value indicating whether the current text's paragraph tag properties are overridden.

# .TextLineInfoGet (VENTURA)

**.TextLineInfoGet** .LineCount=*long*, .LineNumber=*long*, .ColumnNumber=*long*, .Left=*long*, .Top=*long*, .Width=*long*, .Height=*long*, .BaseLineY=*long*

This function returns information about the current paragraph and line. The current paragraph or line is where the insertion point resides.

Syntax	Description
.LineCount	Specifies a numeric variable that is assigned the number of lines in the current paragraph.
.LineNumber	Specifies a numeric variable that is assigned the line number of the current line in the current paragraph.
.ColumnNumber	Specifies a numeric variable that is assigned a value indicating the column the current text is in.
.Left	Specifies the distance from the left side of the current line to the left side of the <u>base page frame</u> , in tenths of a micron. Negative values indicate the left side is to the left of the base page frame.
.Top	Specifies the distance from the top of the current line to the top side of the <u>base page frame</u> , in tenths of a micron. Negative values indicate the top side is above the top of the base page frame.
.Width	Specifies the width of the current line in tenths of a micron.
.Height	Specifies the height of the current line in tenths of a micron.
.BaseLineY	Specifies the distance from the baseline of the current line to the top side of the <u>base page frame</u> , in tenths of a micron. Negative values indicate the baseline is above the top of the base page frame.

## Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

## Example

```
.TextLineInfoGet LineC&, LineN&, , LeftPos&, TopPos&
```

The above example assigns the line count, line number, current line's left and top origin to the **LineC**, **LineN**, **LeftPos**, and **TopPos** variables, respectively.

**.TextSpecialItemType (VENTURA)**

**ReturnValue& = .TextSpecialItemType ( )**

**.TextSpecialItemType .CharCode=*long***

This function returns the type of the special item at the current caret position. It can also be used to return the ANSI number of the character at the current caret position.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies a numeric variable that is assigned a value representing a special item: 0 no special item 1 Attribute 2 Cross reference 3 Date/Time 4 Footnote 5 Endnote 6 Index 7 Index See 8 Index See Also 9 Marker 10 Variable 11 URL 12 Hidden text 13 Anchor 14 Equation 15 Formula 16 Character tag begin 17 Character tag end
<b>.CharCode</b>	Specifies a numeric variable that is assigned the ANSI number of the character at the current caret position.

**Example**

`ReturnValue& = .TextSpecialItemType()`

This example returns the type of the special item at the current caret position.

`.TextSpecialItemType ansinum&`

This example returns the type of the special item at the current caret position.

## .TypeText (VENTURA)

**.TypeText** .Text=*string*

This command inserts text in the publication at the insertion point.

Syntax	Description																																						
<b>.Text</b>	<p>Specifies the text to insert. You can insert special characters such as quotation marks, tabs and returns by using the <b>CHR</b> function.</p> <p>The following list notes some commonly used symbols and special characters, and their ANSI values which are used with the <b>CHR</b> function. These numbers are dependent on the font being used but most standard character fonts use these values:</p> <table><tr><td>34</td><td>Straight double quote</td></tr><tr><td>145</td><td>Typographical open single quote</td></tr><tr><td>146</td><td>Typographical close single quote</td></tr><tr><td>147</td><td>Typographical open double quote</td></tr><tr><td>148</td><td>Typographical close double quote</td></tr><tr><td>150</td><td>En dash</td></tr><tr><td>151</td><td>Em dash</td></tr><tr><td>153</td><td>Trademark symbol</td></tr><tr><td>169</td><td>Copyright symbol</td></tr><tr><td>174</td><td>Registered symbol</td></tr><tr><td>183</td><td>Bullet symbol</td></tr></table> <p>The following list notes some VENTURA text formatting characters and their ANSI values:</p> <table><tr><td>9</td><td>Tab</td></tr><tr><td>10</td><td>Paragraph return</td></tr><tr><td>13</td><td>Forced line break</td></tr><tr><td>17</td><td>Em space</td></tr><tr><td>18</td><td>En space</td></tr><tr><td>19</td><td>Figure space</td></tr><tr><td>20</td><td>Thin space</td></tr><tr><td>21</td><td>Non-breaking space</td></tr></table>	34	Straight double quote	145	Typographical open single quote	146	Typographical close single quote	147	Typographical open double quote	148	Typographical close double quote	150	En dash	151	Em dash	153	Trademark symbol	169	Copyright symbol	174	Registered symbol	183	Bullet symbol	9	Tab	10	Paragraph return	13	Forced line break	17	Em space	18	En space	19	Figure space	20	Thin space	21	Non-breaking space
34	Straight double quote																																						
145	Typographical open single quote																																						
146	Typographical close single quote																																						
147	Typographical open double quote																																						
148	Typographical close double quote																																						
150	En dash																																						
151	Em dash																																						
153	Trademark symbol																																						
169	Copyright symbol																																						
174	Registered symbol																																						
183	Bullet symbol																																						
9	Tab																																						
10	Paragraph return																																						
13	Forced line break																																						
17	Em space																																						
18	En space																																						
19	Figure space																																						
20	Thin space																																						
21	Non-breaking space																																						

### Example

```
.TypeText "Canada"
```

The above example inserts the string **Canada** into a Corel VENTURA publication.

```
.TypeText CHR(9) & CHR(34) & "Canada" & CHR(34)
```

The above example inserts a tab, followed by the string **Canada** in quotation marks into a Corel VENTURA publication.

---

{button ,AL('VENTURA\_TypeText\_Menu;vent\_text;;;',0,"Defaultoverview"),} [Related Topics](#)

## .IsCaretAtEndOfText (VENTURA)

ReturnValue = .IsCaretAtEndOfText ( )

This function returns a value indicating whether the insertion point is at the end of the active text file.

Syntax	Description
ReturnValue	Specifies the numeric variable that is assigned a return value corresponding to whether the insertion point is at the end of the active text file: TRUE (-1) at the end FALSE (0) not at the end

### Note



This command cannot be recorded.



The [.TextEndOfArticle](#) command sends the insertion point to the end of the active text file.

### Example

```
RV = .IsCaretAtEndOfText ( )
```

In the above example, the statement determines whether the insertion point is at the end of the active text (a return value is assigned to **RV**).

---

{button ,AL(^vent\_textnav;;;^,0,"Defaultoverview",)} [Related Topics](#)

## **.PageFirstLine (VENTURA)**

### **.PageFirstLine**

This command sends the insertion point to the beginning of the first text line on the active page. The **.PageFirstLine** command is often used to set VENTURA to text insertion mode. See the example for more information

#### **Note**

**I** The **.PageFirstPara** command also performs the same action.

#### **Example**

```
.FileNew  
.PageFirstLine  
.TypeText "Hello"
```

The above example creates a new file, places the insertion in the first line, and types some text.

---

{button ,AL('vent\_textnav;;;','0',"Defaultoverview"),} [Related Topics](#)

## **.PageFirstPara (VENTURA)**

### **.PageFirstPara**

This command sends the insertion point to the beginning of the first text line on the active page.

#### **Note**



The .PageFirstLine command also performs the same action.

#### **Example**

`.PageFirstPara`

---

`{button ,AL('vent_textnav;;;','0',"Defaultoverview"),}` Related Topics

## **.PageLastLine (VENTURA)**

### **.PageLastLine**

This command sends the insertion point to the beginning of the last text line on the active page.

#### **Note**



The [.PageLastPara](#) command sends the insertion point to the beginning of the last paragraph on the active page.

#### **Example**

`.PageLastLine`

---

`{button ,AL('vent_textnav;;;','0',"Defaultoverview"),}` [Related Topics](#)



## **.PageLastPara (VENTURA)**

### **.PageLastPara**

This command sends the insertion point to the beginning of the last paragraph on the active page.

#### **Note**



The **.PageLastLine** command sends the insertion point to the beginning of the last line on the active page.

#### **Example**

`.PageLastPara`

---

`{button ,AL('vent_textnav;;;','0',"Defaultoverview"),}` **Related Topics**

## .TextCharDown (VENTURA)

**.TextCharDown** .Count=*long*, .Extend=*Boolean*

Moves the insertion point or a selection's active end a specified number of lines down. This command corresponds to using DOWN ARROW or SHIFT+DOWN ARROW in VENTURA.

Syntax	Description
.Count	The number of lines to move down. If not specified, the default is 1.
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified lines from the insertion point; set to FALSE (0) to not select the lines from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect lines. If the active selection was selected from left-to-right, setting to TRUE extends the selection. If the active selection was selected from right-to-left, setting to TRUE deselects lines. Set to FALSE (0) to move the insertion point the specified number of lines from the selection's active end. The default is FALSE.</p>

### Example

```
.TextCharDown 7
```

The above example moves the insertion point down seven lines.

---

{button ,AL('VENTURA\_TextCharDown\_Menu;vent\_textnav;;;',0,"Defaultoverview"),} Related Topics

## .TextCharLeft (VENTURA)

**.TextCharLeft** .Count=*long*, .Extend=*Boolean*

This command moves the insertion point or a selection's active end a specified number of characters positions to the left. This command corresponds to using LEFT ARROW or SHIFT+LEFT ARROW in VENTURA.

Syntax	Description
.Count	The number of character positions to move. If not specified, the default is 1.
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified characters to the left of the insertion point; set to FALSE (0) to not select the characters to the left of the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect characters. If the active selection was selected from left-to-right, setting to TRUE deselects characters. If the active selection was selected from right-to-left, setting to TRUE extends the selection. Set to FALSE (0) to move the insertion point the specified number of characters from the selection's active end. The default is FALSE.</p>

### Example

```
.TextCharLeft 5, -1  
.TextCharRight 3, -1
```

The above example first selects five characters to the left of the insertion point. The second command then de-selects the last three selected characters.

---

{button ,AL("VENTURA\_TextCharLeft\_Menu;vent\_textnav;;;','0,"Defaultoverview"),}} [Related Topics](#)

## .TextCharRight (VENTURA)

**.TextCharRight** .Count=*long*, .Extend=*Boolean*

This command moves the insertion point or a selection's active end a specified number of characters positions to the right. This command corresponds to using RIGHT ARROW or SHIFT+RIGHT ARROW in VENTURA.

Syntax	Description
.Count	The number of character positions to move. If not specified, the default is 1.
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified characters to right of the insertion point; set to FALSE (0) to not select the characters to the right of the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect characters. If the active selection was selected from left-to-right, setting to TRUE extends the selection. If the active selection was selected from right-to-left, setting to TRUE deselects characters. Set to FALSE (0) to move the insertion point the specified number of characters from the selection's active end. The default is FALSE.</p>

### Example

```
.TextCharLeft 5, -1  
.TextCharRight 3, -1
```

The above example first selects five characters to the left of the insertion point. The second command then de-selects the last three selected characters.

---

{button ,AL('VENTURA\_TextCharRight\_Menu;vent\_textnav;;;','0',"Defaulttooverview"),} [Related Topics](#)

## .TextCharUp (VENTURA)

**.TextCharUp** .Count=*long*, .Extend=*Boolean*

Moves the insertion point or a selection's active end a specified number of lines up. This command corresponds to using UP ARROW or the SHIFT+UP ARROW in VENTURA.

Syntax	Description
.Count	The number of lines to move up. If not specified, the default is 1.
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified lines from the insertion point; set to FALSE (0) to not select the lines from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect lines. If the active selection was selected from left-to-right, setting to TRUE deselects lines. If the active selection was selected from right-to-left, setting to TRUE extends the selection. Set to FALSE (0) to move the insertion point the specified number of lines from the selection's active end. The default is FALSE.</p>

### Example

```
.TextCharUp 4
```

The above example moves the insertion point up four lines.

---

{button ,AL('VENTURA\_TextCharUp\_Menu;vent\_textnav;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.TextEndOfArticle (VENTURA)**

**.TextEndOfArticle** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the end of the active text file (article) in VENTURA.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current article from the insertion point; set to FALSE (0) to not select the article from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current article. If the active selection was selected from left-to-right, setting to TRUE extends the selection to the end of the current article. If the active selection was selected from right-to-left, setting to TRUE deselects the current article. Set to FALSE (0) to move the insertion point to the end of the current article. The default is FALSE.</p>

### **Example**

```
.TextStartOfArticle  
.TextEndOfArticle -1
```

The above example moves the insertion point to the beginning of the current article and then to the end, selecting the entire article.

---

{button ,AL('VENTURA\_TextEndOfArticle\_Menu;vent\_textnav;;;','0,"Defaultoverview",')} [Related Topics](#)

## **.TextEndOfColumn (VENTURA)**

**.TextEndOfColumn** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the end of the current text column.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current text column from the insertion point; set to FALSE (0) to not select the text column from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current text column. If the active selection was selected from left-to-right, setting to TRUE extends the selection to the end of the current text column. If the active selection was selected from right-to-left, setting to TRUE deselects the current text column. Set to FALSE (0) to move the insertion point to the end of the current text column. The default is FALSE.</p>

### **Example**

```
.TextStartOfColumn  
.TextEndOfColumn -1
```

The above example moves the insertion point to the beginning of the current text column and then to the end, selecting the entire text column.

---

{button ,AL("VENTURA\_TextEndOfColumn\_Menu;vent\_textnav;;;','0,"Defaultoverview",,)} [Related Topics](#)

## **.TextEndOfFrame (VENTURA)**

**.TextEndOfFrame** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the end of the current text frame.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current text frame from the insertion point; set to FALSE (0) to not select the text frame from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current text frame. If the active selection was selected from left-to-right, setting to TRUE extends the selection to the end of the current text frame. If the active selection was selected from right-to-left, setting to TRUE deselects the current text frame. Set to FALSE (0) to move the insertion point to the end of the current text frame. The default is FALSE.</p>

### **Example**

```
.TextStartOfFrame  
.TextEndOfFrame -1
```

The above example moves the insertion point to the beginning of the current text frame and then to the end, selecting the entire text frame.

---

{button ,AL("VENTURA\_TextEndOfFrame\_Menu;vent\_textnav;;;',0,"Defaultoverview"),} [Related Topics](#)



## **.TextEndOfLine (VENTURA)**

**.TextEndOfLine** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the end of the current line. This command corresponds to using END or the SHIFT+END in VENTURA.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current line from the insertion point; set to FALSE (0) to not select the line from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current line. If the active selection was selected from left-to-right, setting to TRUE extends the selection to the end of the current line. If the active selection was selected from right-to-left, setting to TRUE deselects the current line. Set to FALSE (0) to move the insertion point to the end of the current line. The default is FALSE.</p>

### **Example**

```
.TextStartOfLine  
.TextEndOfLine -1
```

The above example moves the insertion point to the beginning of the current line and then to the end, selecting the entire line.

---

{button ,AL('VENTURA\_TextEndOfLine\_Menu;vent\_textnav;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.TextEndOfPara (VENTURA)**

**.TextEndOfPara** .Extend=*Boolean*

Moves the insertion point or a selection's active end to the end of the current paragraph (after the carriage return).

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current paragraph from the insertion point; set to FALSE (0) to not select the paragraph from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current paragraph. If the active selection was selected from left-to-right, setting to TRUE extends the selection to the end of the paragraph. If the active selection was selected from right-to-left, setting to TRUE deselects the current paragraph. Set to FALSE (0) to move the insertion point to the end of the current paragraph. The default is FALSE.</p>

### **Example**

```
.TextStartOfPara  
.TextEndOfPara -1
```

The above example moves the insertion point to the beginning of the current paragraph and then to the end, selecting the entire paragraph.

---

{button ,AL('VENTURA\_TextEndOfPara\_Menu;vent\_textnav;;;',0,"Defaultoverview",)} [Related Topics](#)

## .TextExtendSelection (VENTURA)

**.TextExtendSelection .ParaAmount=*long*, .CharNumber=*long***

This command selects text.

Syntax	Description
<b>.ParaAmount</b>	Specifies the number of paragraphs to be selected. Specify a positive value to extend forward, and a negative value to extend backward. The default is 0.
<b>.CharNumber</b>	Specifies the character number in the paragraph to select up to. When the <b>ParaAmount</b> value is positive, <b>.CharNumber</b> indicates the position to end in the ending paragraph. When the <b>ParaAmount</b> value is negative, <b>CharNumber</b> indicates the position to begin in the starting paragraph.

### Example

```
.TextExtendSelection 2, 2
```

This example extends the current caret text selection by two paragraphs beginning with the second character in the starting paragraph.

---

{button ,AL('VENTURA\_TextEndOfPara\_Menu;vent\_textnav;;;',0,"Defaultoverview",)} [Related Topics](#)

## .TextFileSelect (VENTURA)

**.TextFileSelect** .FileName=*string*, .ParaNumber=*long*, .CharNumber=*long*, .EndParaNumber=*long*, .EndCharNumber=*long*

This command plants the caret or selects a range of text in the specified text file.

Syntax	Description
.FileName	Specifies the name of the text file to select. If the text file is Export On Save enabled, use the text file name and extension (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties). If the text file is embedded, use the file name which appears in the File List or the VENTURA Navigator (no extension is used). This parameter was modified in Corel VENTURA 8.
.ParaNumber	Specifies the paragraph to move the insertion point to. If not specified, the insertion point moves to the first paragraph.
.CharNumber	Specifies the character to move the insertion point to. If not specified, the insertion point moves to the first character. If the value specified is greater than the number of characters in the paragraph, the insertion point moves to the first character.
.EndParaNumber	Specifies the end paragraph number when selecting the text range. Set to 1 to ignore the parameter; set to 0 to indicate the last paragraph
.EndCharNumber	Specifies the end character number in the end paragraph when selecting the text range. The default is the last character in the paragraph. Specify -1 to ignore the parameter.

### Example

```
.TextFileSelect "c:\BigText.TXT" , 4 , 3
```

The above example moves the insertion point to the third character in the fourth paragraph in the BIGTEXT.TXT file.

---

{button ,AL("VENTURA\_TextFileSelect\_Menu;vent\_textnav;;;",0,"Defaultoverview",)} [Related Topics](#)

## .TextFileSelectGet (VENTURA)

**.TextFileSelectGet .FileName=string , .ParaNumber=long , .CharNumber=long , .EndParaNumber=long , .EndCharNumber=long**

This function returns the current text selection information.

Syntax	Description
.FileName	Specifies the string variable which is assigned the name of text file in which text is selected. If the text file is Export On Save enabled, the text file's full name and extension (this information is available in Corel VENTURA by right-clicking in the VENTURA Navigator and selecting Text File Properties) is assigned. If the text file is embedded, the file name which appears in the File List or the VENTURA Navigator (no extension is used) is assigned.  This parameter was modified in Corel VENTURA 8.
.ParaNumber	Specifies a numeric variable which is assigned the number of the first paragraph of selected text in the selected text file.
.CharNumber	Specifies a numeric variable which is assigned the number of the first character of selected text in the first selected paragraph.
.EndParaNumber	Specifies a numeric variable which is assigned the number of the last paragraph of selected text in the selected text file.
.EndCharNumber	Specifies a numeric variable which is assigned the number of the last character of selected text in the last selected paragraph.

### Example

```
.TextFileSelectGet FileName$, ParaNumber&, CharNumber&
```

This example returns the current text selection information to three variables.

---

{button ,AL('VENTURA\_TextEndOfPara\_Menu;vent\_textnav;;;',0,"Defaultoverview",)} [Related Topics](#)

## .TextParaDown (VENTURA)

**.TextParaDown** .Count=*long*, .Extend=*Boolean*

Moves the insertion point or a selection's active end a specified number of paragraphs down. This command corresponds to using CTRL+DOWN ARROW or the CTRL+SHIFT+DOWN ARROW in VENTURA.

Syntax	Description
.Count	The number of paragraphs to move down. If not specified, the default is 1.
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified paragraphs from the insertion point; set to FALSE (0) to not select the paragraphs from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect paragraphs. If the active selection was selected from left-to-right, setting to TRUE extends the selection. If the active selection was selected from right-to-left, setting to TRUE deselects paragraphs. Set to FALSE (0) to move the insertion point the specified number of paragraphs from the selection's active end. The default is FALSE.</p>

### Note

**I** If the insertion point is in the last paragraph when the command is issued, the insertion point is placed at the end of the paragraph.

### Example

```
.TextParaDown 3, 1
```

The above example extends the selection down three paragraphs.

---

{button ,AL('VENTURA\_TextParaDown\_Menu;vent\_textnav;;;',0,"Defaultoverview"),} Related Topics

## .TextParaUp (VENTURA)

**.TextParaUp** .Count=*long*, .Extend=*Boolean*

Moves the insertion point or a selection's active end a specified number of paragraphs up. This command corresponds to using CTRL+UP ARROW or the CTRL+SHIFT+UP ARROW in VENTURA.

Syntax	Description
.Count	The number of paragraphs to move up. If not specified the default is 1.
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified paragraphs from the insertion point; set to FALSE (0) to not select the paragraphs from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect paragraphs. If the active selection was selected from left-to-right, setting to TRUE deselects paragraphs. If the active selection was selected from right-to-left, setting to TRUE extends the selection. Set to FALSE (0) to move the insertion point the specified number of paragraphs from the selection's active end. The default is FALSE.</p>

### Note

**I** If the insertion point is in the first paragraph when the command is issued, the insertion point is placed at the beginning of the paragraph.

### Example

```
.TextParaUp 5, 1
```

The above example extends the selection up five paragraphs.

```
.TextCharRight 1  
.TextParaUp 1  
.TextParaDown 1, -1
```

The above example selects the entire paragraph. The **TextCharRight** command moves one character in the paragraph to make sure that the insertion point is inside the paragraph. Otherwise, **TextParaUp** could move the insertion point to the beginning of the previous paragraph.

---

{button ,AL('VENTURA\_TextParaUp\_Menu;vent\_textnav;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.TextSelectParagraph (VENTURA)**

### **.TextSelectParagraph**

This command selects the entire paragraph containing the insertion point including the paragraph return character.

#### **Note**



This command corresponds to holding down ALT and clicking in a paragraph.



You can also use the [.TextStartOfPara](#) and the [.TextEndOfPara](#) commands to select a word.

#### **Example**

```
.TextSelectParagraph  
.EditCut
```

The above example selects the paragraph where the insertion point resides and cuts it to the Windows clipboard.

---

{button ,AL('VENTURA\_TextSelectParagraph\_Menu;vent\_textnav;;;','0,"Defaultoverview"),} [Related Topics](#)





## **.TextSelectSentence (VENTURA)**


### **.TextSelectSentence**

This command selects the sentence containing the insertion point. This command considers the period (.), the question mark (?), and the exclamation mark (!) as sentence termination marks. The termination mark is also included in the selection.

#### **Note**

 This command selects all the characters from the sentence termination mark that precedes the insertion point up to, and including, the sentence termination mark that follows the insertion point.

 This command corresponds to holding down CTRL and clicking in a sentence.

 You can also use the .TextSentenceUp and the .TextSentenceDown commands to select a sentence.

#### **Example**

```
.TextStartOfColumn  
.TextSelectSentence
```

The above example moves the insertion point to the beginning of the current text column and then selects the first sentence in the column.

---

{button ,AL("VENTURA\_TextSelectSentence\_Menu;vent\_textnav;;;','0,"Defaultoverview"),} Related Topics

## **.TextSelectWord (VENTURA)**

### **.TextSelectWord**

This command selects the word containing the insertion point. If the insertion point is not in a word (for, example, if it is in a space), the first word that follows the insertion point is selected.

#### **Note**



The spaces following the word are also selected. Punctuation that follows the selected word is not selected.



This command corresponds to double-clicking when editing text in VENTURA.



You can also use the **.TextWordRight** and the **.TextWordLeft** commands to select a word.

#### **Example**

```
.TextStartOfColumn  
.TextSelectWord
```

The above example moves the insertion point to the beginning of the current text column and then selects the first word in the column.

---

{button ,AL('VENTURA\_TextSelectWord\_Menu;vent\_textnav;;;','0,"Defaultoverview",)} **Related Topics**

# .TextSentenceDown (VENTURA)

**.TextSentenceDown** .Count=*long*, .Extend=*Boolean*

Moves the insertion point or a selection's active end a specified number of sentences down.

Syntax	Description
.Count	The number of sentences to move down. If not specified, the default is 1.
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified sentences from the insertion point; set to FALSE (0) to not select the sentences from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect sentences. If the active selection was selected from left-to-right, setting to TRUE extends the selection. If the active selection was selected from right-to-left, setting to TRUE deselects sentences. Set to FALSE (0) to move the insertion point the specified number of sentences from the selection's active end. The default is FALSE.</p>

**Note**

**I** VENTURA considers the punctuation that ends a sentence, part of the sentence. VENTURA also considers each paragraph to have at least one sentence.

**I** If the insertion point is in the last sentence when the command is issued, the insertion point is placed at the end of the sentence.

**Example**

```
.TextSentenceUp 1
.TextSentenceDown 1, -1
```

The above example moves the insertion point to the beginning of the current sentence and then to the end, selecting the entire sentence.

# .TextSentenceUp (VENTURA)

**.TextSentenceUp** .Count=*long*, .Extend=*Boolean*

Moves the insertion point or a selection's active end a specified number of sentences up.

Syntax	Description
.Count	The number of sentences to move up. If not specified, the default is 1.
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified sentences from the insertion point; set to FALSE (0) to not select the sentences from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect sentences. If the active selection was selected from left-to-right, setting to TRUE deselects sentences. If the active selection was selected from right-to-left, setting to TRUE extends the selection. Set to FALSE (0) to move the insertion point the specified number of sentences from the selection's active end. The default is FALSE.</p>

**Note**

**I** VENTURA considers the punctuation that ends a sentence, part of the sentence. VENTURA also considers each paragraph to have at least one sentence.

**I** If the insertion point is in the first sentence when the command is issued, the insertion point is placed at the beginning of the sentence.

**Example**

```
.TextSentenceUp 1
.TextSentenceDown 1, -1
```

The above example moves the insertion point to the beginning of the current sentence and then to the end, selecting the entire sentence.

## .TextStartOfArticle (VENTURA)

**.TextStartOfArticle** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the beginning of the active text file (article) in VENTURA.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current article from the insertion point; set to FALSE (0) to not select the article from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current article. If the active selection was selected from left-to-right, setting to TRUE deselects the current article. If the active selection was selected from right-to-left, setting to TRUE extends the selection to the beginning of the current article. Set to FALSE (0) to move the insertion point to the beginning of the current article. The default is FALSE.</p>

### Example

```
.TextStartOfArticle  
.TextEndOfArticle -1
```

The above example moves the insertion point to the beginning of the current article and then to the end, selecting the entire article.

---

{button ,AL('VENTURA\_TextStartOfArticle\_Menu;vent\_textnav;;;',0,"Defaultoverview"),} [Related Topics](#)

## .TextStartOfColumn (VENTURA)

**.TextStartOfColumn** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the beginning of the current text column.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current text column from the insertion point; set to FALSE (0) to not select the text column from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current text column. If the active selection was selected from left-to-right, setting to TRUE deselects the current text column. If the active selection was selected from right-to-left, setting to TRUE extends the selection to the beginning of the current text column. Set to FALSE (0) to move the insertion point to the beginning of the current text column. The default is FALSE.</p>

### Example

```
.TextStartOfColumn  
.TextEndOfColumn -1
```

The above example moves the insertion point to the beginning of the current text column and then to the end, selecting the entire text column.

---

{button ,AL("VENTURA\_TextStartOfColumn\_Menu;vent\_textnav;;;",0,"Defaultoverview"),} [Related Topics](#)

## .TextStartOfFrame (VENTURA)

**.TextStartOfFrame** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the beginning of the current text frame.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current text frame from the insertion point; set to FALSE (0) to not select the text frame from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current text frame. If the active selection was selected from left-to-right, setting to TRUE deselects the current text frame. If the active selection was selected from right-to-left, setting to TRUE extends the selection to the beginning of the current text frame. Set to FALSE (0) to move the insertion point to the beginning of the current text frame. The default is FALSE.</p>

### Example

```
.TextStartOfFrame  
.TextEndOfFrame -1
```

The above example moves the insertion point to the beginning of the current text frame and then to the end, selecting the entire text frame.

---

{button ,AL("VENTURA\_TextStartOfFrame\_Menu;vent\_textnav;;;','0,"Defaultoverview"),} [Related Topics](#)

## .TextStartOfLine (VENTURA)

**.TextStartOfLine** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the beginning of the current line. This command corresponds to using HOME or SHIFT+HOME in VENTURA.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current line from the insertion point; set to FALSE (0) to not select the line from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current line. If the active selection was selected from left-to-right, setting to TRUE deselects the current line. If the active selection was selected from right-to-left, setting to TRUE extends the selection to the beginning of the current line. Set to FALSE (0) to move the insertion point to the beginning of the current line. The default is FALSE.</p>

### Example

```
.TextStartOfLine  
.TextEndOfLine -1
```

The above example moves the insertion point to the beginning of the current line and then to the end, selecting the entire line.

---

{button ,AL('VENTURA\_TextStartOfLine\_Menu;vent\_textnav;;;','0,"Defaultoverview",')} [Related Topics](#)



## **.TextStartOfPara (VENTURA)**

**.TextStartOfPara** .Extend=*Boolean*

This command moves the insertion point or a selection's active end to the beginning of the current paragraph.

Syntax	Description
.Extend	<p>Specifies whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the current paragraph from the insertion point; set to FALSE (0) to not select the paragraph from the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect the current paragraph. If the active selection was selected from left-to-right, setting to TRUE deselects the current paragraph. If the active selection was selected from right-to-left, setting to TRUE extends the selection to the beginning of the current paragraph. Set to FALSE (0) to move the insertion point to the end of the current paragraph. The default is FALSE.</p>

### **Example**

```
.TextStartOfPara  
.TextEndOfPara -1
```

The above example moves the insertion point to the beginning of the current paragraph and then to the end, selecting the entire paragraph.

---

{button ,AL("VENTURA\_TextStartOfPara\_Menu;vent\_textnav;;;','0,"Defaultoverview"),} [Related Topics](#)

## .TextWordLeft (VENTURA)

**.TextWordLeft** .Count=*long*, .Extend=*Boolean*

Moves the insertion point or a selection's active end a specified number of words positions to the left. This command corresponds to using CTRL+LEFT ARROW or CTRL+SHIFT+LEFT ARROW in VENTURA.

Syntax	Description
.Count	The number of words to move over. If not specified, the default is 1.
.Extend	<p>Specifies the whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified words to the left of the insertion point; set to FALSE (0) to not select the words to the left of the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect words. If the active selection was selected from left-to-right, setting to TRUE deselects words. If the active selection was selected from right-to-left, setting to TRUE extends the selection. Set to FALSE (0) to move the insertion point the specified number of words from the selection's active end. The default is FALSE.</p>

### Note

**I** VENTURA considers a single space that precedes a word as part of the word. VENTURA also considers punctuation marks, tab characters, line return characters, and paragraph return characters as words. For example, VENTURA considers the sentence **The "gray" fox is mine.** as eight words (three punctuation marks and five words).

### Example

```
.TextWordRight 6, -1
.TextWordLeft 2, -1
```

The above example first selects six words to the right of the insertion point. The second command then de-selects the last two selected words.

---

{button ,AL("VENTURA\_TextWordLeft\_Menu;vent\_textnav;;;",0,"Defaultoverview"),} [Related Topics](#)

## .TextWordRight (VENTURA)

**.TextWordRight** .Count=*long*, .Extend=*Boolean*

Moves the insertion point or a selection's active end a specified number of words positions to the right. This command corresponds to using CTRL+RIGHT ARROW or CTRL+SHIFT+RIGHT ARROW in VENTURA.

Syntax	Description
.Count	The number of words to move over. If not specified, the default is 1.
.Extend	<p>Specifies the whether to select text or change a selection's active end.</p> <p>If text is not already selected, set to TRUE (-1) to select the specified words to the right of the insertion point; set to FALSE (0) to not select the words to the right of the insertion point. The default is FALSE.</p> <p>If text is already selected, set to TRUE (-1) to extend the selection or to deselect words. If the active selection was selected from left-to-right, setting to TRUE extends the selection. If the active selection was selected from right-to-left, setting to TRUE deselects words. Set to FALSE (0) to move the insertion point the specified number of words from the selection's active end. The default is FALSE.</p>

### Note

**I** VENTURA considers a single space that precedes a word as part of the word. VENTURA also considers punctuation marks, tab characters, line return characters, and paragraph return characters as words. For example, VENTURA considers the sentence **The "gray" fox is mine.** as eight words (three punctuation marks and five words).

### Example

```
.TextWordRight 6, -1
.TextWordLeft 2, -1
```

The above example first selects six words to the right of the insertion point. The second command then de-selects the last two selected words.




---

{button ,AL('VENTURA\_TextWordRight\_Menu;vent\_textnav;;;',0,"Defaultoverview"),} [Related Topics](#)


## .FormatIndex (VENTURA)

**.FormatIndex** **.IndexName=string**, **.Prefix=string**, **.Suffix=string**, **.NumberFormat=string**, **.See=string**, **.SeeAlso=string**, **.Delimiter=string**, **.LetterHeading=Boolean**, **.Categories=string**, **.ChapterName=string**, **.FileName=string**

This command sets the properties for an index in the active document.

Syntax	Description
<b>.IndexName</b>	Specifies the name of the index to set properties for. If this index doesn't exist, it is created.
<b>.Prefix</b>	Specifies the text that appears before the page number.
<b>.Suffix</b>	Specifies the text that appears after the page number.
<b>.NumberFormat</b>	Specifies the numbering format. Use the string "[C#]" to indicate a chapter number. Use the string "[P#]" to indicate a page number. For example, to create a format which uses the word "Chapter" followed by the chapter number, a hyphen, and a page number, use the string "Chapter [C#]-[P#]".
<b>.See</b>	Specifies the text prefix for cross-reference entries which direct readers to other terms in an index.
<b>.SeeAlso</b>	Specifies the text prefix for cross-reference entries which direct readers to other terms in an index.
<b>.Delimiter</b>	<p>Specifies the character(s) used to separate the index numbers in entries that occur on more than one page. The character(s) are specified as a string. For example, to use a slash, set the parameter to "/".</p> <p>This string can also use VENTURA markup codes. For example, to set the slash to bold, use the string "&lt;B&gt;/&lt;D&gt;". For more information about VENTURA markup codes, click .</p>
<b>.LetterHeading</b>	Specifies whether to enable the Letter Heading option. The letter heading option places an A before entries starting with A, a B before entries starting with B, and so on. Set to TRUE (-1) to enable this option; otherwise, set to FALSE (0).
<b>.Categories</b>	<p>Specifies the categories to use in a segregated index as a string. For example, if four categories exist (Cat1, Cat2, Cat3, and Cat4), and only three were required, the string to specify the categories would have the following format:</p> <p>"Cat1,Cat2,Cat4"</p> <p>The string is case-sensitive and only commas can be used to separate the categories (no spaces).</p> <p>To generate a non-segregated index  one that includes entries from all categories</p>
 set the string to "Default".	
<b>.ChapterName</b>	Specify the chapter in which to add the index text file. If the chapter does not exist, it is created.
<b>.FileName</b>	Specifies the name for the index text file.

### Note

 This command corresponds to the Properties command in the VENTURA Navigator. Click Tools, Navigator, and choose Index from the list box. Right-click, then click Properties from the menu.

### Example

```
.FormatIndex "Parts", , , "Chp [C#]-Pg [P#]", "See", "See Also", "/", TRUE, "Default",  
"Chapter1", "Parts"
```

The above example sets the properties for the **Parts** index.

---

{button ,AL(vent\_toc;;;,0,"Defaultoverview"),} [Related Topics](#)

## **.FormatIndexCount (VENTURA)**

**ReturnValue& = .FormatIndexCount ( )**

This function returns the number of indexes in the active document.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the number of indexes.

### **Note**



This command cannot be recorded.

### **Example**

```
Icounts& = .FormatIndexCount ( )
```

The number of indexes in the active document is assigned to the **Icounts** variable.

---

{button ,AL(^;vent\_toc;;;','0,"Defaultoverview",)} [Related Topics](#)


## **.FormatIndexDelete (VENTURA)**

**.FormatIndexDelete** .IndexName=*string*

This command deletes a specified index from the active document.

Syntax	Description
<b>.IndexName</b>	Specifies the name of the index to delete. It must exist in the active document.

### **Note**

 This command corresponds to the Delete command in the VENTURA Navigator. Click Tools, Navigator, and choose Index from the list box. Choose an index, right-click, then click Delete from the menu.

### **Example**

```
.FormatIndexDelete "Old_Index"
```

The above example deletes the index named Old\_Index from the active document.




---

`{button ,AL(';vent_toc;;;','0,"Defaultoverview",)}` [Related Topics](#)



# .FormatIndexGet (VENTURA)

**.FormatIndexGet** **.IndexName=string**, **.Prefix=string**, **.Suffix=string**, **.NumberFormat=string**, **.See=string**, **.SeeAlso=string**, **.Delimiter=string**, **.LetterHeading=Boolean**, **.Categories=string**, **.ChapterName=string**, **.FileName=string**

This function returns the index properties for a specified index.

Syntax	Description
<b>.IndexName</b>	Specifies the name of the index in the active document to obtain properties for.
<b>.Prefix</b>	Specifies the string variable that is assigned the text that appears before the page number.
<b>.Suffix</b>	Specifies the string variable that is assigned the text that appears after the page number.
<b>.NumberFormat</b>	Specifies the string variable that is assigned the numbering format.
<b>.See</b>	Specifies the string variable that is assigned the text prefix for cross-reference entries. These direct readers to other terms in an index.
<b>.SeeAlso</b>	Specifies the string variable that is assigned the text prefix for cross-reference entries. These direct readers to other terms in an index.
<b>.Delimiter</b>	Specifies the string variable that is assigned the character(s) used to separate the index numbers in entries that occur on more than one page. The assigned string can also use VENTURA markup codes used to format the characters. For more information about VENTURA markup codes, click  .
<b>.LetterHeading</b>	Specifies the variable that is assigned the Letter Heading option. The letter heading option places an A before entries starting with A, a B before entries starting with B, and so on. If enabled, TRUE (-1) is assigned; otherwise, FALSE (0) is assigned.
<b>.Categories</b>	Specifies the string variable that is assigned the categories in a segregated index. For example, if more than one category is used, the assigned string would have the following format: "Cat1,Cat2,Cat3" Commas are used to separate the categories.  If the index is non-segregated  one that includes entries from all categories
	 the assigned string is "Default".
<b>.ChapterName</b>	Specifies the string variable that is assigned the index text file's chapter.
<b>.FileName</b>	Specifies the string variable that is assigned the index text file's filename.

## Note

-  The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.
-  This command cannot be recorded.

## Example

```
LastIndex& = .FormatIndexCount ( ) 'number of indexes
LastIndexName$ = .FormatIndexGetAt (LastIndex&)
.FormatIndexGet LastIndexName$, , , NF$, SeeText$, SeeAlsoText$, Idelim$, LH
```

In the above example, the first statement counts the number of indexes. The second statement is assigned the name of the last created index in the active document. In the last line, the function returns the last index's properties to the **NF\$**, **SeeText\$**, **SeeAlsoText\$**, **Idelim&**, and **LH** variables.

## **.FormatIndexGetAt (VENTURA)**

**ReturnString\$ = .FormatIndexGetAt .Index=*long***

This function returns the index name associated with an index's index number.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of the index.
<b>.Index</b>	Specifies the index number of an index. Index numbers are not associated with index order in the Navigator (alphabetical order). Instead, index numbers are associated by the order in which the indexes are created. The first index created is 1, the second index created is 2, and so on. The last created index is equal to the return value from the <a href="#"><b>.FormatIndexCount</b></a> function. If an index is deleted, the index numbers are recompiled.

### **Note**



This command cannot be recorded.

### **Example**

```
LastIndex& = .FormatIndexCount ( ) 'number of indexes  
LastIndexName$ = .FormatIndexGetAt (LastIndex&)
```

In the above example, the first statement counts the number of indexes. The second statement is assigned the name of the last created index in the active document.

---

**{button ,AL(';vent\_toc;;;','0,"Defaultoverview",)} [Related Topics](#)**




## **.FormatIndexRename (VENTURA)**

**.FormatIndexRename** *.IndexName=string, .NewName=string*

This command renames an existing index in the active document.

Syntax	Description
<b>.IndexName</b>	Specifies the name of the index to rename.
<b>.NewName</b>	Specifies the new name for the specified index. An index with this name cannot currently exist in the active document.

### **Note**

 This command corresponds to the Rename command in the VENTURA Navigator. Click Tools, Navigator, and choose Index from the list box. Right-click, then click Rename from the menu.

### **Example**

```
.FormatIndexRename "Old_Index", "New_Index"
```

The above example renames the index named Old\_Index to New\_Index.

---

`{button ,AL(;vent_toc;;;,0,"Defaultoverview",)}` [Related Topics](#)

## **.FormatIndexUpdate (VENTURA)**

**.FormatIndexUpdate .IndexName=string**

This command generates or updates an index in the active document.

Syntax	Description
<b>.IndexName</b>	Specifies the index to generate an index text file for. This index must already exist in the active document.

### **Note**

**I** This command corresponds to the Update command in the VENTURA Navigator. Select an index in the VENTURA Navigator, right-click, and click Update.

### **Example**

```
LastIndex& = .FormatIndexCount ( )      'number of indexes
LastIndexName$ = .FormatIndexGetAt (LastIndex&)
.FormatIndexUpdate LastIndexName$
```

In the above example, the first statement counts the number of indexes. The second statement is assigned the name of last created index in the active document. In the last line, the **.FormatIndexUpdate** command generates an index for the last index in the document.

---

{button ,AL(';vent\_toc;;;',0,"Defaultoverview",)} Related Topics


## .FormatTOC (VENTURA)

**.FormatTOC** *.TocName=string, .NumOfLevels=long, .ChapterName=string, .FileName=string*

This command sets the properties for a table of contents in the active document.

Syntax	Description
<b>.TocName</b>	Specifies the name of the table of contents to set properties for. If the table of contents does not exist, it is created.
<b>.NumOfLevels</b>	Specifies the number of levels to appear in the table of contents. The levels are determined by paragraph tags assigned to the headings in the document. The maximum number of levels is 10.
<b>.ChapterName</b>	Specifies the string variable that is assigned the table of contents text file's chapter.
<b>.FileName</b>	Specifies the string variable that is assigned the table of contents text file's filename.

### Note

 This command corresponds to the Properties command in the VENTURA Navigator. Click Tools, Navigator, and choose Table of Contents from the list box. Right-click, then click Properties from the menu.

### Example

```
.FormatTOC "Business Plan", 4
.FormatTOCLevelSet "Business Plan", 1, "Main Heading", "Chapter [C#]"
.FormatTOCLevelSet "Business Plan", 2, "Minor Heading", , "Page [P#]"
```

The above example, the first line sets the number of levels in the Business Plan table of contents to 4. The second line sets properties for the first level and the third line sets the properties for the second level.

---

{button ,AL('VENTURA\_FormatTableOfContent\_Menu;vent\_toc;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.FormatTOCCount (VENTURA)**

**ReturnValue& = .FormatTOCCount ( )**

This function returns the number of table of contents' in the active document.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the number of table of contents'.

### **Note**



This command cannot be recorded.

### **Example**

```
TOCcounts& = .FormatTOCCount ( )
```

The number of table of contents' in the active document is assigned to the TOCcounts variable.

---

`{button ,AL("VENTURA_TableOfContentCount_Menu;vent_toc;;;','0,"Defaultoverview"),}` [Related Topics](#)


## **.FormatTOCDelete (VENTURA)**

**.FormatTOCDelete .TocName=string**

This command deletes a specified table of contents from the active document.

Syntax	Description
<b>.TocName</b>	Specifies the name of the table of contents to delete. It must exist in the active document.

### **Note**

 This command corresponds to the Delete command in the VENTURA Navigator. Click Tools, Navigator, and choose Table of Contents from the list box. Choose a table of contents, right-click, then click Delete from the menu.

### **Example**

`.FormatTOCDelete "Old_Toc"`

The above example deletes the table of contents named Old\_Toc from the active document.

---

`{button ,AL('VENTURA_DeleteTableOfContent_Menu;vent_toc;;;','0,"Defaultoverview",)} Related Topics`

## .FormatTOCGet (VENTURA)

**.FormatTOCGet** **.TocName**=*string*, **.NumOfLevels**=*long*, **.ChapterName**=*string*, **.FileName**=*string*

This function returns the table of contents properties for a specified table of contents.

Syntax	Description
<b>.TocName</b>	Specifies the name of the table of contents to obtain properties for.
<b>.NumOfLevels</b>	Specifies the numeric variable that is assigned the number of levels in the specified table of contents.
<b>.ChapterName</b>	Specifies the string variable that is assigned the table of contents text file's chapter.
<b>.FileName</b>	Specifies the string variable that is assigned the table of contents text file's filename.

### Note



The variable specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatTOC "Business Plan", BPLevels&
```

The above example assigns the number of levels in the Business Plan table of contents to the **BPLevels** variable.

---

{button ,AL("VENTURA\_FormatTableOfContentGet\_Menu;vent\_toc;;;",0,"Defaultoverview"),} Related Topics

## **.FormatTOCGetAt (VENTURA)**

**ReturnString\$ = .FormatTOCGetAt .TocIndex=*long***

This function returns the table of contents name associated with a table of contents index number in the active document.

Syntax	Description
<b>ReturnString\$</b>	Specifies the string variable that is assigned the name of the table of contents.
<b>.TocIndex</b>	Specifies the index number of a table of contents. Index numbers are not associated with the table of contents order in the Navigator (alphabetical order). Instead, index numbers are associated by the order in which the table of contents are created. The first table of contents created is 1, the second table of contents created is 2, and so on. The last created table of contents is equal to the return value from the <b><u>.FormatTOCCount</u></b> function. If a table of contents is deleted, the index numbers are recompiled.

### **Note**

 This command cannot be recorded.

### **Example**

```
LastTOC& = .FormatTOCCount ( )      'number of tags  
LastTOCName$ = .FormatTOCGetAt (LastTOC&)
```

In the above example, the first statement counts the number of table of contents'. The second statement is assigned the name of the last created table of contents in the active document.

---

**{button ,AL("VENTURA\_TableOfContentGetAt\_Menu;vent\_toc;;;','0,"Defaultoverview",)} Related Topics**

## .FormatTOCLevelGetAt (VENTURA)

**.FormatTOCLevelGetAt** TocName=*string*, .Level=*long*, .ParagraphTag=*string*, .Prefix=*string*, .Suffix=*string*

This function returns the properties of a level in an existing table of contents in the active document.

Syntax	Description
<b>.TocName</b>	Specifies the name of the table of contents to obtain level properties for.
<b>.Level</b>	Specifies the level to obtain properties for.
<b>.ParagraphTag</b>	Specifies the string variable that is assigned the paragraph tag that is used for the specified level in the table of contents.
<b>.Prefix</b>	Specifies the string variable that is assigned the text to insert before each specified level in the table of contents.
<b>.Suffix</b>	Specifies the string variable that is assigned the text to insert after each specified level in the table of contents.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.FormatTOCLevelGetAt "Business Plan", 2, ParaTag$, BP_Prefix$, BP_Suffix$
```

In the above example, the properties for the second level of the Business Plan table of contents are assigned to **ParaTag**, **BP\_Prefix**, and **BP\_Suffix**.

---

{button ,AL("VENTURA\_FormatTocLevelGetAt\_Menu;vent\_toc;;;','0,"Defaultoverview"),} Related Topics




# .FormatTOCLevelSet (VENTURA)

**.FormatTOCLevelSet .TocName=string, .Level=long, .ParagraphTag=string, .Prefix=string, .Suffix=string**

This command sets the properties for a level in an existing table of contents in the active document.

Syntax	Description
.TocName	Specifies the name of the table of contents to set level properties for.
.Level	Specifies the level to set properties for. The level must already exist in the table of contents. Use the <a href="#">.FormatTOC</a> command to set the number of levels.
.ParagraphTag	Specifies the paragraph tag that is used for the specified level in the table of contents.
.Prefix	Specifies the text to insert before each specified level in the table of contents. The following codes can be used to specify special numbering formats: [C#] Chapter number [P#] Paragraph number [S#] Section number
.Suffix	Specifies the text to insert after each specified level in the table of contents. The following codes can be used to specify special numbering formats: [C#] Chapter number [P#] Paragraph number [S#] Section number

### Note

 This command corresponds to the Properties command in the VENTURA Navigator. Click Tools, Navigator, and choose Table of Contents from the list box. Right-click, then click Properties from the menu.

### Example

```
.FormatTOC "Business Plan", 4
.FormatTOCLevelSet "Business Plan", 1, "Main Heading", "Chapter [C#]"
.FormatTOCLevelSet "Business Plan", 2, "Minor Heading", , "Page [P#]"
```

In the above example, the first line sets the number of levels in the Business Plan table of contents to 4. The second line sets properties for the first level and the third line sets the properties for the second level.


## **.FormatTOCRename (VENTURA)**

**.FormatTOCRename .TocName=*string*, .NewName=*string***

This command renames an existing table of contents in the active document.

<b>Syntax</b>	<b>Description</b>
<b>.TocName</b>	Specifies the name of the table of contents to rename.
<b>.NewName</b>	Specifies the new name for the specified table of contents. A table of contents with this name cannot currently exist in the active document.

### **Note**

 This command corresponds to the Rename command in the VENTURA Navigator. Click Tools, Navigator, and choose Table of Contents from the list box. Choose a table of contents, right-click, then click Rename from the menu.

### **Example**

`.FormatTOCRename "Old_Toc", "New_TOC"`

The above example renames the table of contents named Old\_Toc to New\_Toc.

---

`{button ,AL('VENTURA_RenameTableOfContent_Menu;vent_toc;;;','0,"Defaultoverview",)} Related Topics`


## **.FormatTOCUpdate (VENTURA)**

**.FormatTOCUpdate .TocName=string**

This command generates or updates a table of contents in the active document.

Syntax	Description
<b>.TocName</b>	Specifies the table of contents for which a text file will be generated. This table of contents must already exist in the active document.

### **Note**

 This command corresponds to the Update command in the VENTURA Navigator. Select a table of contents in the VENTURA Navigator, right-click, and click Update.

### **Example**

```
LastTOC& = .FormatTOCCount ( )      'number of tags
LastTOCName$ = .FormatTOCGetAt (LastTOC&)
.FormatTOCUpdate LastTOCName$
```

In the above example, the first statement counts the number of table of contents'. The second statement is assigned the name of the last created table of contents in the active document. In the last line, the **.FormatTOCUpdate** command generates a table of contents for the last table of contents in the document.

---

**{button ,AL(';vent\_toc;;;','0,"Defaultoverview",)} Related Topics**

## .IndexEntryGet (VENTURA)

**.IndexEntryGet** .Category=*string* , .Type=*long* , .Levels=*long*

This function returns the properties of an active index entry.

Syntax	Description
.Category	Specifies a string variable which is assigned the category of the index entry.
.Type	Specifies a numeric variable which is assigned the index entry type: 0     Index 1     See 2     See also
.Levels	Specifies a numeric variable which is assigned the levels in the index entry.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This function was introduced in Corel VENTURA 8.

### Example

```
.IndexEntryGet ICat, IType, ILevels
```

The above example returns index information to three string variables.

---

{button ,AL(';vent\_toc;;;','0,"Defaultoverview",)} Related Topics

## .IndexEntryGetAt (VENTURA)

**.IndexEntryGetAt** *.Level=long*, *.Entry=string* , *.SortKey=string*

This function returns the level properties of an active index entry.

Syntax	Description
<b>.Level</b>	Specifies the level to return properties for.
<b>.Entry</b>	Specifies a string variable which is assigned the index text.
<b>.SortKey</b>	Specifies a string variable which is assigned the sort key.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This function was introduced in Corel VENTURA 8.

### Example

```
.IndexEntryGetAt 2, IEntry$, IKey$
```

The above example returns index level (second level) information to the **IEntry** and **IKey** string variables.

---


`{button ,AL(';vent_toc;;;','0,"Defaultoverview",)} Related Topics`


# .InsertIndexAddLevel (VENTURA)

`.InsertIndexAddLevel .Entry=string, .SortKey=string`


This command, in conjunction with the [.InsertIndexBegin](#) and [.InsertIndexEnd](#) commands, inserts an index entry at the insertion point.

To insert an index entry, the [.InsertIndexBegin](#) command must first be used. This command is then followed by one or more [.InsetIndexAddLevel](#) commands. The last [.InsetIndexAddLevel](#) command is then followed by the [.InsertIndexEnd](#) command. The first (and required) [.InsetIndexAddLevel](#) command after the [.InsertIndexBegin](#) command sets the main index entry. The second (and optional) [.InsetIndexAddLevel](#) command sets an index subentry. The third [.InsetIndexAddLevel](#) command sets the second index subentry, and so on.

Syntax	Description
<code>.Entry</code>	Specifies the index entry text.
<code>.SortKey</code>	Specifies where the entry is to appear in the index. Use <b>.SortKey</b> to change the usual order of certain entries  such as those beginning with numbers

 For example, select "N" to have 1984 appear under "N" in the index rather than the exclamation mark heading under which non-alphabetic entries are usually listed.

### Note

 This command corresponds to the Insert Index Entry dialog box. Click Insert, Index Entry while the insertion point is placed in a word.

### Example

See the [.InsertIndexBegin](#) command for an example.

---



`{button ,AL(;vent_toc;;;','0,"Defaultoverview",)} Related Topics`

# **.InsertIndexBegin (VENTURA)**


**.InsertIndexBegin** .Category=*string*, .Type=*long*

This command, in conjunction with the [.InsertIndexAddLevel](#) and [.InsertIndexEnd](#) commands, inserts an index entry at the insertion point.

To insert an index entry, the **.InsertIndexBegin** command must first be used. This command is then followed by one or more **.InsetIndexAddLevel** commands. The last **.InsetIndexAddLevel** is then followed by the **.InsertIndexEnd** command.

Syntax	Description
.Category	Specifies the index category of the index entry for segregated indexes. If this category does not exist, it is created.  To use the index entry in a non-segregated index  one that includes entries from all categories   set this parameter to "Default". If this parameter is omitted, it is set to "Default".
.Type	Specifies the index type for the index entry: 0      Index (default if omitted) 1      See 2      See also

## **Note**

 This command corresponds to the Insert Index Entry dialog box. Click Insert, Index Entry while the insertion point is placed in a word.

## **Example**

```
.TypeText "bruins"  
.InsertIndexBegin "Default", 0  
    .InsertIndexAddLevel "bears"  
    .InsertIndexAddLevel "other names"  
.InsertIndexEnd
```

The above example inserts the text **bruins** into the active document. An index entry is then inserted with the **bears** as the main entry, and **other names** as a subentry.

## **.InsertIndexEnd (VENTURA)**

### **.InsertIndexEnd**

This command, in conjunction with the [.InsertIndexAddLevel](#) and [.InsertIndexBegin](#) commands, inserts an index entry at the insertion point.

To insert an index entry, the [.InsertIndexBegin](#) command must first be used. This command is then followed by one or more [.InsetIndexAddLevel](#) commands. The last [.InsetIndexAddLevel](#) command is then followed by the [.InsertIndexEnd](#) command which closes the index entry procedure.

#### **Note**



This command corresponds to the Insert Index Entry dialog box. Click Insert, Index Entry while the insertion point is placed in a word.

#### **Example**

See the [.InsertIndexBegin](#) command for an example.

---

`{button ,AL(';vent_toc;;;','0,"Defaultoverview",)} Related Topics`



## .ToolsSpellCheckBegin (VENTURA)

**ReturnValue& = .ToolsSpellCheckBegin .Word=string**

This function returns the number of suggestions the Spell Checker suggests for a specified misspelled word. After this function is used in a script, the **.ToolsSpellCheckEnd** command should be used to reset the Spell Checker.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the number of suggestions the Spell Checker suggests for a misspelled word.
<b>.Word</b>	Specifies the word to be spell checked. If the word contains a space, for example, "dog " instead of "dog", the Spell Checker provides suggestions. Remove spaces with the <b>LTRIM</b> or <b>RTRIM</b> Corel SCRIPT functions. See the example for more information.

### Note

**I** This command cannot be recorded.

### Example

```
DO WHILE NOT .IsCaretAtEndOfText() 'continue looping until the end of text
    .TextSelectWord 'select the next word
    Word$ = .SelectedText() 'assign selected word
    Word$ = RTRIM(Word$) 'remove spaces from word
    SuggestCount& = .ToolsSpellCheckBegin(Word$) 'test word's spelling
    Index% = 1 'sets a counter
    'the following loop is used to display a message box of suggested words
    'for misspelled words
    DO WHILE Index% <= SuggestCount&
        Suggest$ = .ToolsSpellSuggestionGetAt(Index%)
        MESSAGE Word & ": Not found" & CHR(13) & "Suggestion: " & Suggest$
        Index% = Index% + 1
        Suggest$ = .ToolsSpellSuggestionGetAt(Index%)
    LOOP
    .TextCharRight 1 'moves the insertion point a character to the right
LOOP
.ToolsSpellCheckEnd 'resets spell checking
```

The above example begins spell checking words from the insertion point to the end of the text. See the script comments for an explanation of this example.

---

{button ,AL('venttools;;;',0,"Defaultoverview",)} [Related Topics](#)

## **.ToolsSpellCheckEnd (VENTURA)**

### **.ToolsSpellCheckEnd**

After using the [.ToolsSpellCheckBegin](#) function, use the **.ToolsSpellCheckEnd** command to reset the Spell Checker. Resetting the Spell Checker sets the value in memory returned by the [.ToolsSpellCheckBegin](#) function to 0.

#### **Note**



This command cannot be recorded.

#### **Example**

See the example in the [.ToolsSpellCheckBegin](#) function.

---

`{button ,AL('venttools;;;',0,"Defaultoverview"),}` [Related Topics](#)

## **.ToolsSpellSuggestionGetAt (VENTURA)**

**ReturnString\$ = .ToolsSpellSuggestionGetAt .SpellIndex=*long***

This function returns a spelling suggestion for a misspelled word. This function must follow a [\*\*.ToolsSpellCheckBegin\*\*](#) function in a script that has returned a value greater than 0 (which indicates that it has found a misspelled word).

Syntax	Description
<b>ReturnString\$</b>	Specifies a string variable that is assigned a spelling suggestion for a misspelled word.
<b>.SpellIndex</b>	Specifies the index number of a spelling suggestion. Index numbers range in value from 1 to the value returned by the previous instance of a <a href="#"><b>.ToolsSpellCheckBegin</b></a> function in a script. The first index number corresponds to the first spelling suggestion, the second index number corresponds to the second spelling suggestion, and so on.

### **Note**



This function cannot be recorded.



This function must precede the [\*\*.ToolsSpellCheckEnd\*\*](#) command which is used to reset the Spell Checker.

### **Example**

See the example in the [\*\*.ToolsSpellCheckBegin\*\*](#) function.

---

**{button ,AL("venttools;;;",0,"Defaultoverview",)} [Related Topics](#)**

## **.ToolsThesaurusBegin (VENTURA)**

**ReturnValue& = .ToolsThesaurusBegin .Word=string**

This function returns the number of definitions the Thesaurus suggests for a specified word. After this function is used in a script, the **.ToolsThesaurusEnd** command should be used to reset the Thesaurus.

Syntax	Description
<b>ReturnValue&amp;</b>	Specifies the numeric variable that is assigned the number of definitions the Thesaurus suggests for a specified word.
<b>.Word</b>	Specifies the word to be looked up.

### **Note**

- 1** If the word is not found in the Thesaurus' dictionary, 0 is returned.
- 1** This command cannot be recorded.

### **Example**

```
DIM Word AS STRING, Definition AS STRING 'declares variables
DIM DefCount AS LONG, DefIndex AS LONG 'declares variables
DIM Count AS LONG, Index AS LONG 'declares variables

DO WHILE NOT .IsCaretAtEndOfText() 'continue looping until the end of text
    .TextSelectWord 'select the next word
    Word$ = .SelectedText() 'assign selected word
    Word$ = RTRIM(Word$) 'remove spaces from word
    DefCount = .ToolsThesaurusBegin(Word$) 'counts the number of definitions for word
    DefIndex = 1
    'the following loop is used to display message boxes
    'with selected word definition and alternatives
    DO WHILE DefIndex <= DefCount
        .ToolsThesaurusDefinitionGet, DefIndex, Definition, Count 'gets a definition
        message "Definition: " & Word$ & CHR(13) & Definition
        Index = 1
        DO WHILE Index <= Count
            Alt = .ToolsThesaurusGetAt(DefIndex, Index) 'returns an alternative word
            message "Alternate words for " & Word$ & CHR(13) & Alt
            Index = Index + 1
        LOOP
        DefIndex = DefIndex+1
    LOOP
    .TextCharRight 1 'moves the insertion point a character to the right
LOOP
.ToolsThesaurusEnd 'resets Thesaurus
```

The above example starts the Thesaurus the insertion point until the end of the text. See the script comments for an explanation of this example.

---

{button ,AL('venttools;;;','0,"Defaultoverview",)} [Related Topics](#)

## **.ToolsThesaurusDefinitionGet (VENTURA)**

**.ToolsThesaurusDefinitionGet .DefIndex=*long*, .Definition=*string*, .NumOfAlts=*long***

This function returns the definition and the number of alternatives associated with the previously executed [.ToolsThesaurusBegin](#) function.

Syntax	Description
<b>.DefIndex</b>	Specifies the definition index number. Index numbers range in value from 1 to the value returned by the previous instance of a <a href="#">.ToolsThesaurusBegin</a> function in a script. The first index number corresponds to the first definition, the second index number corresponds to the second definition, and so on.
<b>.Definition</b>	Specifies the string variable that is assigned the definition of the word associated to the index number specified in the <b>.DefIndex</b> parameter.
<b>.NumOfAlts</b>	Specifies the numeric variable that is assigned the number of alternatives associated to the index number specified in the <b>.DefIndex</b> parameter.

### **Note**



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### **Example**

See the example in the [.ToolsThesaurusBegin](#) function.

---

**{button ,AL('venttools;;;',0,"Defaultoverview"),}** [Related Topics](#)

## **.ToolsThesaurusEnd (VENTURA)**

### **.ToolsThesaurusEnd**

After using the [.ToolsThesaurusBegin](#) function to return the number of definitions the Thesaurus suggests for a specified word, use the **.ToolsThesaurusEnd** command to reset the Thesaurus. The value returned by the [.ToolsThesaurusBegin](#) function is set to 0 when the Thesaurus is reset.

#### **Note**



This command cannot be recorded.

#### **Example**

See the example in the [.ToolsThesaurusBegin](#) function.

---

{button ,AL('venttools;;;','0',"Defaultoverview",)} [Related Topics](#)

## **.ToolsThesaurusGetAt (VENTURA)**

**ReturnString\$ = .ToolsThesaurusGetAt .DefIndex=*long*, .AltIndex=*long***

This function returns an alternative word based on a specified definition and alternative item number. This command must be preceded in script by a [\*\*.ToolsThesaurusBegin\*\*](#) command.

Syntax	Description
<b>ReturnString\$</b>	Specifies a string variable that is assigned an alternative word based on a specified definition and alternative item number.
<b>.DefIndex</b>	Specifies the definition index number. Index numbers range in value from 1 to the value returned by the previous instance of a <a href="#"><b>.ToolsThesaurusBegin</b></a> function in a script. The first index number corresponds to the first definition, the second index number corresponds to the second definition, and so on.
<b>.AltIndex</b>	Specifies the alternative word index number. Alternative word index number range in value from 1 to the value returned by the previous instance of a <a href="#"><b>.ToolsThesaurusDefinitionGet</b></a> function ( <b>.NumOfAlts</b> parameter) in a script. The first index number corresponds to the first alternative word, the second index number corresponds to the second alternative word, and so on.

### **Note**

 This function cannot be recorded.

### **Example**

See the example in the [\*\*.ToolsThesaurusBegin\*\*](#) function.

---

**{button ,AL("venttools;;;",0,"Defaultoverview"),}** [Related Topics](#)

## **.CurrentView (VENTURA)**

**ReturnValue& = .CurrentView ( )**

This function returns the active view in Corel VENTURA.

<b>Syntax</b>	<b>Description</b>
<b>ReturnValue&amp;</b>	Specifies the current view type of the current publication. 0     Page layout 1     Page tag (formerly named Master page view) 2     Copy editor

### **Example**

```
MyView& = .CurrentView( )
```

This example returns the active view type to the variable MyView.

---

{button ,AL('VENTURA\_View\_Menu;;;;','0',"Defaultoverview",)} [Related Topics](#)



## .SetVisible (VENTURA)

**.SetVisible** **.Show=Boolean**, **.Activate=Boolean**

This command makes the Corel VENTURA application visible or hidden on your Windows desktop. When VENTURA is hidden, it runs in the Windows background and is not visible on screen. Running as a hidden application can make a VENTURA script run faster since it does not have to redraw a publication when changes are applied to it.

Syntax	Description
<b>.Show</b>	Specifies whether VENTURA is visible or hidden. Set to TRUE (-1) to show the Corel VENTURA application. Set to FALSE (0) to hide the application.
<b>.Activate</b>	Specifies whether to VENTURA the active application in Windows. Set to TRUE (-1) to enable this option; otherwise set to FALSE (0) which is the default if omitted.

### Note



By clicking CTRL+ALT+DELETE, you can find all the applications Windows is running, both visible and hidden.



This command is not recordable.



If you execute a script that calls VENTURA from the Corel SCRIPT Editor, VENTURA becomes visible if the **.SetVisible** command is enabled. See [Executing application commands in the background](#) for more information.

### Example

```
.SetVisible -1
```

The above example shows the Corel VENTURA application.

---

{button ,AL('VENTURA\_SetVisible\_Menu;vent\_view;;;','0','Defaultoverview',)} [Related Topics](#)

## .ViewColorCorrection (VENTURA)

**.ViewColorCorrection .CorrectionType=long**, .SimulatePrinter=*Boolean*, .GamutAlarm=*Boolean*, .ApplyToText=*Boolean*,

In Page Layout or Page tag (formerly Master page) mode, this command sets VENTURA's color correction. Color correction allows you to see colors as they will appear when printed.

Syntax	Description
<b>.CorrectionType</b>	Specifies the correction type to apply: 1 None 2 Fast (general approximation) 3 Accurate (best screen colors)
.SimulatePrinter	Specifies whether to enable or disable the display of colors in the publication to match as closely as possible the colors in the printed output. Set to TRUE (-1) to enable; otherwise, set to FALSE (0). This option can only be enabled if <b>.CorrectionType</b> is set to 2 or 3.
.GamutAlarm	Specifies whether to enable or disable the display colors in the publication which cannot be reproduced by the selected printer. Such colors display as a single solid color specified using the Options command in the Tools menu. Set to TRUE (-1) to enable; otherwise, set to FALSE (0). This option can only be enabled if <b>.SimulatePrinter</b> is set to TRUE.
.ApplyToText	Specifies whether color correction applies to all elements in the active publication, or to text only. Set to TRUE (-1) to apply to text only. Set to FALSE (0) to apply to all elements. This option is only used if <b>.CorrectionType</b> is set to 2 or 3.

### Note



This command cannot be recorded.



This command corresponds to the Color Correction command on the View menu in Corel VENTURA. Click View, Color Correction.

### Note



This command cannot be recorded.

### Example

```
.ViewPageLayout  
.ViewColorCorrection 2, -1, -1
```

The above example sets the color correction to fast and enables printer simulation. The Gamut alarm is also enabled.

---

{button ,AL('VENTURA\_View\_Menu;vent\_view;;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.ViewCopyEditor (VENTURA)**

### **.ViewCopyEditor**

This command sets the view of the active publication in VENTURA to Copy Editor mode. This mode can speed up text editing by displaying text without text formatting, and by not displaying graphics and frames.

#### **Note**



This command corresponds to the Copy Editor command in the View menu in Corel VENTURA. Click View, Copy Editor.



You can also use the [ViewPageLayout](#) or [ViewMasterPage](#) commands to set the view in VENTURA.

#### **Example**

`.ViewCopyEditor`

The above example sets the view of the active publication to Copy Editor mode.

---

`{button ,AL('VENTURA_View_Menu;vent_view;;;','0','Defaultoverview'),}` [Related Topics](#)




# .ViewFrameBorders (VENTURA)

**.ViewFrameBorders** .Visible=*Boolean*

In Page Layout or Page tag (formerly Master page) mode, this command shows or hides the frame borders in the active VENTURA publication.

Syntax	Description
.Visible	Specifies whether to display frame borders. Set to TRUE (-1) to show the frame borders. Set to FALSE (0) to hide the frame borders. If omitted, the setting is TRUE.

**Note**

-  This command cannot be recorded.
-  This command corresponds to the Frame Borders command in the View menu in Corel VENTURA. Click View, Frame Borders.
-  Wireframe borders always display and print for frames in Draft view, even when the Hide Frame Borders command in the View menus is enabled.

**Example**

```
.ViewPageLayout
.ViewFrameBorders -1
```

The above example displays the frame borders in the active VENTURA publication.

---

{button ,AL('VENTURA\_View\_Menu;vent\_view;;;','0',"Defaultoverview"),} Related Topics

## **.ViewFreezeScreen (VENTURA)**

### **.ViewFreezeScreen**

This command prevents the screen from redrawing during the execution of a script for VENTURA. This command can be used to speed up script execution without excessive screen redrawing.

#### **Note**



The termination of a VENTURA script restores screen redrawing.



You can restore screen redrawing by using the [.ViewUnFreezeScreen](#) command.

#### **Example**

```
.ViewFreezeScreen
```

---

```
{button ,AL(^VENTURA_View_Menu;vent_view;;;',0,"Defaultoverview"),} Related Topics
```


# .ViewGotoMasterPage (VENTURA)

**.ViewGotoMasterPage** .PageName=*string*, .RightPage=*Boolean*, .GoWhere=*long*

In Page tag (formerly Master page) view, this command moves the insertion point to a specified Page tag (formerly Master page) in the active publication.

Syntax	Description
.PageName	Specifies the name of the Page tag (formerly Master page) to go to. This parameter is only used if the <b>.GoWhere</b> parameter is set to 4.
.RightPage	Specifies whether to send the insertion point to the left or right side Page tag (formerly Master page). Set to TRUE (-1) to go to the right side; otherwise, set to FALSE (0) to go to the left side which is the default if omitted.
.GoWhere	Specifies the Page tag (formerly Master page) to send the insertion point to: 0 First page 1 Last page 2 Previous page 3 Next page 4 Specified page in the <b>.PageName</b> parameter. If this parameter is omitted, it is set to 4.

### Note

 This command corresponds to the Go To command in the View menu in Corel VENTURA. Click View, Go to.

### Example

```
.ViewMasterPage
.ViewGotoMasterPage "Intro", FALSE, 4
```

The above example sets the view of the active publication to Page tag (formerly Master page) mode. Once in Page tag (formerly Master page) mode, the insertion point is sent to the Intro Page tag (formerly Master page).


---

{button ,AL('VENTURA\_ViewGotoPage\_Menu;vent\_view;;;',0,"Defaultoverview",)} Related Topics


# .ViewGotoPage (VENTURA)

**.ViewGotoPage** .PageNumber=*long*, .ChapterOrFile=*string*, .RelativeTo=*long*, .GoWhere=*long*, .UserNumbering=*Boolean*

This command moves the insertion point to a specified page in the active publication.

Syntax	Description
.PageNumber	Specifies the number of the page to display. To use this parameter, set <b>.GoWhere</b> to 4.
.ChapterOrFile	Specifies the name of the chapter or text file to go to.
.RelativeTo	Specifies the relative page number: 0     relative to chapter 1     relative to text 2     relative to publication (default setting)
.GoWhere	Specifies the page to go to: 0     first page 1     last page 2     previous page 3     next page 4     specified page (see the <b>.PageNumber</b> parameter) 5     Previous chapter 6     Next chapter
.UserNumbering	Specifies whether to use user page numbers or VENTURA page numbers. Set to TRUE (-1) to use user page numbering. Set to FALSE (0) to use VENTURA page numbering.  User page numbers are those that appear in headers or footers. Ventura page numbers coincide with the physical location of a page within the publication. Ventura page numbers differ from the user page numbers when numbering is restarted at some point in the publication.  for example, at the beginning of each chapter.

## Note

 This command corresponds to the Go to command in the View menu in Corel VENTURA. Click View, Go to.

## Example

`.ViewGotoPage 4, "Chapter1", 0, 4, FALSE`

The above example sends the insertion point to the fourth page in Chapter1.

---

{button ,AL('VENTURA\_ViewGotoPage\_Menu;vent\_view;;;','0,"Defaultoverview",')} Related Topics


## **.ViewGraphicResolution (VENTURA)**


**.ViewGraphicResolution .Resolution=*Boolean***

In Page Layout or Page tag (formerly Master page) mode, this command sets the graphic resolution display. Changing resolution affects only the way graphics are displayed on screen, not print preview or printing resolution.

Syntax	Description
<b>.Resolution</b>	Specifies whether to set High or Low resolution. Set to TRUE (-1) to use high resolution. Set to FALSE (0) to use low resolution. If omitted, the setting is toggled.

### **Note**

 Enabling High Resolution maximizes display resolution in the publication area, but decreases screen redraw speed. Differences in the speed of display are most noticeable with complex graphics. Enabling Low Resolution displays graphics as low-resolution bitmaps, which speeds up initial import of files, reopening of publications, and screen redraws, when you move around in your publication.

 With High Resolution selected, Corel vector formats such as CDR (from CorelDRAW), display using the actual vector image. With Low Resolution selected, the image header displays instead of the vector image. If no image header is available, VENTURA creates one.

### **Example**

```
.ViewPageLayout
.ViewGraphicResolution -1
```

The above example sets the graphic resolution to high.

---

{button ,AL('VENTURA\_View\_Menu;vent\_view;;;','0',"Defaultoverview"),} [Related Topics](#)



## .ViewHideAllGraphics (VENTURA)

**.ViewHideAllGraphics** .Hide=*Boolean*

In Page Layout or Page tag (formerly Master page) mode, this command shows or hides the graphics in the active VENTURA publication. Hiding graphics speeds up screen redrawing.

Syntax	Description
.Hide	Specifies whether to display graphics. Set to TRUE (-1) to hide the graphics. Set to FALSE (0) to show the graphics. If omitted, the setting is TRUE.

### Note



This command cannot be recorded.



This command corresponds to the Hide Pictures command in the View menu in Corel VENTURA. Click View, Hide Pictures.



Hidden graphics are replaced with rectangles containing an X across.

### Example

```
.ViewHideAllGraphics -1
```

The above example hides all graphics.

---

{button ,AL('VENTURA\_View\_Menu;vent\_view;;;','0',"Defaultoverview"),} [Related Topics](#)

## **.ViewMasterPage (VENTURA)**

### **.ViewMasterPage**

This command sets the view of the active publication in Corel VENTURA to Page tag view (or Master Page view in pre-Corel VENTURA 8 versions).

#### **Note**



This command corresponds to the Page tag command in the View menu in Corel VENTURA. Click View, Page Tag.



You can also use the [ViewPageLayout](#) or [ViewCopyEditor](#) commands to set the view in VENTURA.

#### **Example**

`.ViewMasterPage`

The above example sets the view of the active publication to Page tag (formerly Master page) mode.

---

`{button ,AL('VENTURA_View_Menu;vent_view;;;','0','Defaultoverview'),}` [Related Topics](#)

# .ViewPageFooter (VENTURA)

**.ViewPageFooter** .Visible=*Boolean*

In Page Layout or Page tag (formerly Master page) mode, this command shows or hides the footer in the active VENTURA publication.

Syntax	Description
.Visible	Specifies whether to display footers. Set to TRUE (-1) to show the footers. Set to FALSE (0) to hide the footers. If omitted, the setting is TRUE.

**Note**



This command cannot be recorded.



This command corresponds to the Page Footer command in the Page menu in Corel VENTURA. Click Page, Show Footer.

**Example**

`.ViewPageFooter -1`

The above example displays the footers in the active VENTURA publication.

---

`{button ,AL('VENTURA_View_Menu;vent_view;;;','0','Defaultoverview'),}` [Related Topics](#)

## .ViewPageFootnote (VENTURA)


**.ViewPageFootnote** .Visible=*Boolean*

In Page Layout or Page tag (formerly Master page) mode, this command shows or hides the footnotes in the active Corel VENTURA publication.

This command is obsolete in Corel VENTURA 8.

Syntax	Description
.Visible	Specifies whether to display footnotes. Set to TRUE (-1) to show the footnotes. Set to FALSE (0) to hide the footnotes. If omitted, the setting is TRUE.

### Note

 This command cannot be recorded.

### Example

```
.ViewPageFootnote -1
```

The above example displays the footnotes in the active Corel VENTURA publication.

---

{button ,AL("VENTURA\_View\_Menu;vent\_view;;;','0,"Defaultoverview"),} [Related Topics](#)

# .ViewPageHeader (VENTURA)

**.ViewPageHeader** .Visible=*Boolean*

In Page Layout or Page tag (formerly Master page) mode, this command shows or hides the header in the active VENTURA publication.

Syntax	Description
.Visible	Specifies whether to display headers. Set to TRUE (-1) to show the headers. Set to FALSE (0) to hide the headers. If omitted, the setting is toggled.

**Note**



This command cannot be recorded.



This command corresponds to the Page Header command in the Page menu in Corel VENTURA. Click Page, Show Header.

**Example**

`.ViewPageHeader -1`

The above example displays the headers in the active VENTURA publication.

---

`{button ,AL('VENTURA_View_Menu;vent_view;;;','0','Defaultoverview'),}` [Related Topics](#)

## **.ViewPageLayout (VENTURA)**

### **.ViewPageLayout**

This command sets the view of the active publication in VENTURA to Page Layout mode. Page Layout mode, the default setting, displays the publication the way it looks when it is printed, including the text layout, formatting, and graphics.

#### **Note**



This command corresponds to the Page Layout command in the View menu in Corel VENTURA. Click View, Page Layout.



You can also use the [ViewCopyEditor](#) or [ViewMasterPage](#) commands to set the view in VENTURA.

#### **Example**

`.ViewPageLayout`

The above example sets the view of the active publication to Page Layout mode.

---

`{button ,AL('VENTURA_View_Menu;vent_view;;;','0',"Defaultoverview"),}` [Related Topics](#)

## **.ViewRedraw (VENTURA)**

### **.ViewRedraw**

This command refreshes the entire publication on screen, removing screen dirt left by earlier operations.

#### **Note**



This command cannot be recorded.



This command corresponds to clicking Window, Refresh Window in Corel VENTURA.

#### **Example**

`.ViewRedraw`

The above command refreshes the active publication.

---

`{button ,AL('VENTURA_ViewRedraw_Menu;vent_view;;;','0,"Defaultoverview",)} Related Topics`

## .ViewRulers (VENTURA)

**.ViewRulers** *.Visible=Boolean*

This command shows or hides the horizontal and vertical rulers.

Syntax	Description
.Visible	Specifies whether to display the rulers. Set to TRUE (-1) to show the rulers. Set to FALSE (0) to hide the rulers. If omitted, the setting is TRUE.

### Note



This command cannot be recorded.



This command corresponds to the Rulers command in the View menu in Corel VENTURA. Click View, Rulers.

### Example

```
.ViewRulers -1
```

The above example displays the rulers in VENTURA.

---

{button ,AL("VENTURA\_View\_Menu;vent\_view;;;",0,"Defaultoverview",)} Related Topics



## **.ViewUnFreezeScreen (VENTURA)**

### **.ViewUnFreezeScreen**

This command restores screen redrawing that has been disabled by the [.ViewFreezeScreen](#) command.

#### **Example**

```
.ViewFreezeScreen  
.DrawRectangle 123123, 225000, 444000, 123123  
.DrawEllipse 250000, 333000, 500000, 222444  
.ViewUnFreezeScreen
```

The above example freezes screen redrawing, performs two graphic commands, and unfreezes the screen.

---

{button ,AL('VENTURA\_View\_Menu;vent\_view;;;',0,"Defaultoverview"),} [Related Topics](#)

## **.ViewZoom (VENTURA)**

**.ViewZoom .ZoomPercent=long**

In Page Layout or Page tag (formerly Master page) mode, this command controls the magnification (zoom) of the active VENTURA publication.

Syntax	Description
<b>.ZoomPercent</b>	Specifies the magnification factor for your publication relative to its default (100). Valid zoom percentages range from 10 to 1250.

### **Note**



This command corresponds to the Zoom command in the View menu in Corel VENTURA. Click View, Zoom.

### **Example**

`.ViewZoom 200`

The above example changes the publication magnification to 200.

---

`{button ,AL("VENTURA_View_Menu;vent_view;;;','0,"Defaultoverview",,)} Related Topics`

## **.ViewZoomLayout (VENTURA)**

**.ViewZoomLayout .LayoutType=*long*, .NumberOfPages=*long***

In Page Layout or Page tag (formerly Master page) mode, this command controls page layout view of the active VENTURA publication.

Syntax	Description
<b>.LayoutType</b>	Specifies the page layout view: <div><div>1</div>Page Width</div> <div><div>2</div>Page Height</div> <div><div>3</div>Full Page</div> <div><div>4</div>Facing Pages</div>
<b>.NumberOfPages</b>	Specifies the number of pages to display.

### **Note**



This command corresponds to the Zoom command in the View menu in Corel VENTURA. Click View, Zoom.

### **Example**

`.ViewZoomLayout 1`

The above example changes the diagram magnification, so that the full width of the page of the active publication is displayed.

---

`{button ,AL("VENTURA_View_Menu;vent_view;;;','0,"Defaultoverview"),}` [Related Topics](#)


## **.AppWindowPosition (VENTURA)**

**.AppWindowPosition .Left=*long*, Top=*long*, Width=*long*, Height=*long***

This command sets the size and position of the Corel VENTURA window on your desktop.

<b>Syntax</b>	<b>Description</b>
<b>.Left</b>	Specifies the distance in screen pixels from the VENTURA window's left border to the left side of the monitor's display area.
<b>.Top</b>	Specifies the distance in screen pixels from VENTURA window's top border to the top side of the monitor's display area.
<b>.Width</b>	Specifies the VENTURA window's width in screen pixels.
<b>.Height</b>	Specifies the VENTURA window's height in screen pixels.

### **Note**

 This command cannot be recorded.

### **Example**

```
.AppWindowPosition 50, 75, 500, 425
```

The above example repositions and sets the dimensions of the VENTURA window.

---

{button ,AL('vent\_windows;;;;',0,"Defaultoverview",)} [Related Topics](#)

## .AppWindowPositionGet (VENTURA)

**.AppWindowPositionGet** .Left=*long*, Top=*long*, Width=*long*, Height=*long*

This functions returns the size and position of the Corel VENTURA window on your desktop.

Syntax	Description
.Left	Specifies the numeric variable that is assigned the distance in screen pixels from the VENTURA window's left border to the left side of the monitor's display area.
.Top	Specifies the numeric variable that is assigned the distance in screen pixels from VENTURA window's top border to the top side of the monitor's display area.
.Width	Specifies the numeric variable that is assigned the VENTURA window's width in screen pixels.
.Height	Specifies the numeric variable that is assigned the VENTURA window's height in screen pixels.

### Note



The variables specified in this function must be explicitly declared, or implicitly declared using a type-declaration suffix.



This command cannot be recorded.

### Example

```
.AppWindowPosition VWinLeft&, VWinTop&, VWinWidth&, VWinHeight&
```

The above example returns the properties of the VENTURA window to the variables **VWinLeft**, **VWinTop**, **VWinWidth**, and **VWinHeight**.

---

{button ,AL('vent\_windows;;;;',0,"Defaultoverview",)} [Related Topics](#)

## .CountWindows (VENTURA)

**ReturnValue**& = .CountWindows ( )

This function returns the number of open document windows in the current VENTURA session. This number corresponds to the number of open documents in the Window menu in VENTURA.

Syntax	Description
<b>ReturnValue</b> &	The numeric variable that is assigned the number of open documents in the current VENTURA sessions.

### Note



This command cannot be recorded.

### Example

```
DocCount& = .CountWindows ( )
```

The above example returns the number of open documents in the current VENTURA session to the variable **DocCount**.

---

{button ,AL("Ventrura\_CountWindows\_Menu;vent\_windows;;;','0,"Defaultoverview"),} Related Topics

## .NextDocWindow (VENTURA)

**.NextDocWindow** .DocIndex=*long*

This command activates the document window that immediately follows the active document window in the VENTURA Window menu. This command can also be used to activate a specified document.

Syntax	Description
.DocIndex	Specifies the document window number to open. The numbers are listed on the Window menu in VENTURA. If this parameter is not specified, the document window that immediately follows the active document window on the VENTURA Window menu is activated.

### Note



This command is ignored if there is only one document window.



If **.DocIndex** is not specified, and the active document window is the last on the Window menu list when the command is issued, the first document window is activated.



See also the **.PrevDocWindow** command.

### Examples

.NextDocWindow 3

The above example activates the third listed document window on the Window menu in VENTURA.

---

{button ,AL('VENTURA\_NextDocWindow\_Menu;vent\_windows;;;','0,"Defaultoverview",')} Related Topics

## **.PrevDocWindow (VENTURA)**

### **.PrevDocWindow**

This command activates the document window that immediately precedes the active document window in the VENTURA Window menu.

#### **Note**



This command is ignored if there is only one document window.



If the active document window is the first in the Window menu list when the command is issued, the last document window is activated.



See also the [.NextDocWindow](#) command.

#### **Example**

```
.FileOpen "C:\annual report.vp"  
.FileOpen "D:\analysis.vp"  
.PrevDocWindow
```

The above example opens two documents, and then activates the first document which was opened.

---




{button ,AL("VENTURA\_PrevDocWindow\_Menu;vent\_windows;;;','0,"Defaultoverview"),} [Related Topics](#)



## .ToolsAutoScript (VENTURA)




**.ToolsAutoScript** .Enable=*Boolean*

This command enables and disables auto script execution during script execution of File menu commands that use auto scripts. The following commands may trigger an auto script:

-  **.FileNew**
-  **.FileOpen**
-  **.FileClose**
-  **.FilePrint**
-  **.FileSave**
-  **.FileExit**

Syntax	Description
.Enable	Specifies whether to enable or disable VENTURA auto scripts during script execution. To enable auto scripts, set to TRUE (-1); to disable set to FALSE (0). If omitted, sets to FALSE.

### Note

-  The **.ToolsAutoScript** command cannot be used to disable the **OnStart** auto script.
-  See [Auto Scripts](#) for more information.
-  This command cannot be recorded.

### Example

```
.ToolsAutoScript 0
.FileNew
```

The above example prevents the OnNew auto script from executing (if exists) when the **.FileNew** command is executed.

