

Antenna House PDF SDK V6 / PDF Viewer Ctrl インタフェース仕様

## 目次

---

Antenna House PDF SDK V6 / PDF Viewer Ctrl インタフェース仕様.....	1
1. 概要 .....	1
1.1. PDFPageViewCtrl .....	1
1.2. PDFThumbnailCtrl .....	4
1.3. PDFOutlineCtrl .....	4
1.4. PDFLayerCtrl .....	5
2. 動作環境 .....	6
2.1. 対応 OS .....	6
2.2. 仮想環境 .....	6
3. 制限事項 .....	6
3.1. 対象 PDF .....	6
3.2. コントロール全般 .....	6
3.3. レンダリング品質および性能について .....	7
3.4. 複数プロセス/スレッドでの同時使用 .....	8
4. コントロール情報 .....	9
4.1. .NET/.NET Framework 版 .....	9
4.2. 動作に必要なモジュール、手続き等 .....	10
4.3. 64bit コントロール .....	12
4.4. 座標系について .....	14
4.5. タッチ操作 .....	17
4.6. PDF 保存時の暗号化 .....	17
4.7. 文字列検索 .....	17
4.8. オブジェクト ID .....	18
4.9. ファイルオープン時の一時ファイルへのコピー .....	18
5. PDFPageViewCtrl .....	20
5.1. 概要 .....	20
5.2. チュートリアル .NET Framework 版 .....	20
5.3. アプリケーションが実装すべき項目 .....	21
5.4. API 一覧 .....	23
5.4.1. プロパティ .....	23

5.4.2.	メソッド .....	25
5.4.3.	イベント .....	27
<b>5.5.</b>	<b>API リファレンス .....</b>	<b>29</b>
5.5.1.	IsOpen プロパティ .....	29
5.5.2.	IsModified プロパティ .....	29
5.5.3.	IsLockedByOwnerPassword プロパティ .....	29
5.5.4.	IsSearched プロパティ .....	29
5.5.5.	IsSearchedAll プロパティ .....	29
5.5.6.	HasFocus プロパティ .....	30
5.5.7.	PageCount プロパティ .....	30
5.5.8.	ShowCoverPageDuringTwoUp プロパティ .....	30
5.5.9.	PageShadow プロパティ .....	30
5.5.10.	BackgroundColor プロパティ .....	30
5.5.11.	TextSelectedColor プロパティ .....	31
5.5.12.	SearchedAllColor プロパティ .....	31
5.5.13.	HilightTextColor プロパティ .....	31
5.5.14.	PageLayout プロパティ .....	31
5.5.15.	ViewMode プロパティ .....	32
5.5.16.	SelectMode プロパティ .....	32
5.5.17.	Direction プロパティ .....	34
5.5.18.	CtrlStyle プロパティ .....	34
5.5.19.	HasUserPassword プロパティ .....	36
5.5.20.	HasOwnerPassword プロパティ .....	36
5.5.21.	UserPassword プロパティ .....	36
5.5.22.	OwnerPassword プロパティ .....	36
5.5.23.	SecurityPermissionFlag プロパティ .....	37
5.5.24.	ZoomRatio プロパティ .....	37
5.5.25.	ZoomRatioLimit プロパティ .....	37
5.5.26.	CurrentPageNumber プロパティ .....	38
5.5.27.	CanUndo プロパティ .....	38
5.5.28.	CanRedo プロパティ .....	38
5.5.29.	CanCut プロパティ .....	38

5.5.30.	CanCopy プロパティ .....	38
5.5.31.	CanPaste プロパティ .....	39
5.5.32.	CanDelete プロパティ .....	39
5.5.33.	CanModifyPage プロパティ .....	39
5.5.34.	CanModifyAnnotation プロパティ .....	39
5.5.35.	CanModifyFontStyle プロパティ .....	39
5.5.36.	CanUndoView プロパティ .....	40
5.5.37.	CanRedoView プロパティ .....	40
5.5.38.	CanPrint プロパティ .....	40
5.5.39.	SelectedObjectCount プロパティ .....	40
5.5.40.	PdfVersion プロパティ .....	40
5.5.41.	CreationDate プロパティ .....	41
5.5.42.	ModifyDate プロパティ .....	41
5.5.43.	Title プロパティ .....	41
5.5.44.	Author プロパティ .....	41
5.5.45.	Subject プロパティ .....	41
5.5.46.	Keywords プロパティ .....	42
5.5.47.	Creator プロパティ .....	42
5.5.48.	Producer プロパティ .....	42
5.5.49.	DocumentPath プロパティ .....	42
5.5.50.	SDKVersion プロパティ .....	42
5.5.51.	SearchStartPage プロパティ .....	43
5.5.52.	LoadPageCount プロパティ .....	43
5.5.53.	SelectedTextString プロパティ .....	43
5.5.54.	ViewCoord プロパティ .....	43
5.5.55.	IncludeArtifact プロパティ .....	44
5.5.56.	SnapPointType プロパティ .....	44
5.5.57.	HasRedactAnnotation プロパティ .....	44
5.5.58.	ScrollPosition プロパティ .....	45
5.5.59.	DeviceScrollPosition プロパティ .....	45
5.5.60.	OpenDocument メソッド .....	45
5.5.61.	SaveDocument メソッド .....	45

5.5.62.	CloseDocument メソッド .....	46
5.5.63.	PrintDocument メソッド .....	46
5.5.64.	GetPDFDocumentHandle メソッド .....	50
5.5.65.	AttachPDFDocument メソッド .....	50
5.5.66.	UnlockOwnerPasswordSecurity メソッド .....	50
5.5.67.	GetPageWidth メソッド .....	51
5.5.68.	GetPageHeight メソッド .....	51
5.5.69.	GetNextZoomRatio メソッド .....	51
5.5.70.	GotoPage メソッド .....	52
5.5.71.	GotoFirstPage メソッド .....	52
5.5.72.	GotoLastPage メソッド .....	52
5.5.73.	GotoPreviousPage メソッド .....	52
5.5.74.	GotoNextPage メソッド .....	52
5.5.75.	GetPageRotation メソッド .....	53
5.5.76.	RotatePages メソッド .....	53
5.5.77.	ResetView メソッド .....	53
5.5.78.	Undo メソッド .....	53
5.5.79.	Redo メソッド .....	54
5.5.80.	Cut メソッド .....	54
5.5.81.	Copy メソッド .....	54
5.5.82.	Paste メソッド .....	54
5.5.83.	Delete メソッド .....	54
5.5.84.	SelectAll メソッド .....	55
5.5.85.	ShowDocumentPropertyDialog メソッド .....	55
5.5.86.	ShowFontStyleDialog メソッド .....	55
5.5.87.	ShowObjectPropertyDialog メソッド .....	55
5.5.88.	UndoView メソッド .....	55
5.5.89.	RedoView メソッド .....	56
5.5.90.	ShowSearchDialog メソッド .....	56
5.5.91.	Search メソッド .....	56
5.5.92.	SearchNext メソッド .....	57
5.5.93.	SearchPrevious メソッド .....	57

5.5.94. ClearSelect メソッド .....	57
5.5.95. MakeBmpPage メソッド .....	57
5.5.96. MakeJpegPage メソッド .....	58
5.5.97. MakePngPage メソッド .....	59
5.5.98. MakeTiffPage メソッド .....	59
5.5.99. MakeSvgPage メソッド .....	60
5.5.100. GetDefaultObjectProperty/ GetDefaultObjectProperty2 メソッド .....	60
5.5.101. SetDefaultObjectProperty/ SetDefaultObjectProperty2 メソッド .....	61
5.5.102. GetCursorPosition メソッド .....	64
5.5.103. ScrollToPosition メソッド .....	64
5.5.104. ScrollToDevicePosition メソッド .....	65
5.5.105. ClientToLogical メソッド .....	65
5.5.106. SetSelectModeEx メソッド .....	65
5.5.107. CreateObject メソッド .....	66
5.5.108. GetObjectCount メソッド .....	71
5.5.109. GetObjectInfo メソッド .....	71
5.5.110. DeleteObjects メソッド .....	71
5.5.111. ModifyObjects メソッド .....	72
5.5.112. HitTest メソッド .....	72
5.5.113. GetNextObjectId メソッド .....	73
5.5.114. SetHilightText メソッド .....	73
5.5.115. ClearHilightText メソッド .....	74
5.5.116. GetTextInfo メソッド .....	74
5.5.117. GetTextInfoInRect メソッド .....	78
5.5.118. ClientToLogicalRect メソッド .....	78
5.5.119. SetCtrlCursor メソッド .....	79
5.5.120. SetSmoothing メソッド .....	79
5.5.121. SelectTextString メソッド .....	79
5.5.122. SetFocus メソッド .....	80
5.5.123. GetSelectedTextInfo メソッド .....	80
5.5.124. FitViewToRect メソッド .....	80
5.5.125. GetLicenseInfo メソッド .....	81

5.5.126. ShowMeasureDialog メソッド .....	81
5.5.127. RedactAndSaveDocument メソッド .....	82
5.5.128. DocumentOpened イベント .....	82
5.5.129. PageTurned イベント .....	83
5.5.130. Zoom イベント .....	83
5.5.131. DragDropFiles イベント .....	83
5.5.132. ObjectAdded イベント .....	83
5.5.133. DoAction イベント .....	83
5.5.134. DefaultObjectPropertyChanged イベント .....	84
5.5.135. MouseDownC イベント .....	84
5.5.136. MouseUpC イベント .....	85
5.5.137. DoubleClickC イベント .....	85
5.5.138. MouseMoveC イベント .....	85
5.5.139. ContextMenuC イベント .....	86
5.5.140. KeyDownC イベント .....	86
5.5.141. KeyUpC イベント .....	86
5.5.142. ObjectDeleted イベント .....	86
5.5.143. ObjectResized イベント .....	87
5.5.144. ObjectMoved イベント .....	87
5.5.145. EditBoxShowed イベント .....	87
5.5.146. EditBoxTextEdited イベント .....	87
5.5.147. Scrolled イベント .....	88
<b>6. PDFThumbnailCtrl .....</b>	<b>89</b>
<b>6.1. 概要 .....</b>	<b>89</b>
<b>6.2. チュートリアル .NET Framework 版 .....</b>	<b>89</b>
<b>6.3. アプリケーションが実装すべき項目 .....</b>	<b>90</b>
<b>6.4. API 一覧 .....</b>	<b>92</b>
6.4.1. プロパティ .....	92
6.4.2. メソッド .....	93
6.4.3. イベント .....	93
<b>6.5. API リファレンス .....</b>	<b>95</b>
6.5.1. IsOpen プロパティ .....	95

6.5.2.	IsModified プロパティ .....	95
6.5.3.	IsLockedByOwnerPassword プロパティ .....	95
6.5.4.	HasFocus プロパティ .....	95
6.5.5.	CurrentPageNumber プロパティ .....	95
6.5.6.	PageCount プロパティ .....	96
6.5.7.	SelectedPageCount プロパティ .....	96
6.5.8.	CtrlStyle プロパティ .....	96
6.5.9.	BackgroundColor プロパティ .....	96
6.5.10.	TextColor プロパティ .....	97
6.5.11.	ThumbnailSize プロパティ .....	97
6.5.12.	CanUndo プロパティ .....	97
6.5.13.	CanRedo プロパティ .....	97
6.5.14.	CanCut プロパティ .....	97
6.5.15.	CanCopy プロパティ .....	98
6.5.16.	CanPaste プロパティ .....	98
6.5.17.	CanDelete プロパティ .....	98
6.5.18.	CanModifyPage プロパティ .....	98
6.5.19.	HasUserPassword プロパティ .....	98
6.5.20.	HasOwnerPassword プロパティ .....	99
6.5.21.	UserPassword プロパティ .....	99
6.5.22.	OwnerPassword プロパティ .....	99
6.5.23.	SecurityPermissionFlag プロパティ .....	99
6.5.24.	DocumentPath プロパティ .....	100
6.5.25.	SDKVersion プロパティ .....	100
6.5.26.	SelectedPages プロパティ .....	100
6.5.27.	OpenDocument メソッド .....	101
6.5.28.	SaveDocument メソッド .....	101
6.5.29.	CloseDocument メソッド .....	101
6.5.30.	GetPDFDocumentHandle メソッド .....	101
6.5.31.	AttachPDFDocument メソッド .....	102
6.5.32.	UnlockOwnerPasswordSecurity メソッド .....	102
6.5.33.	GotoPage メソッド .....	102

6.5.34.	GetPageRotation メソッド .....	103
6.5.35.	RotatePages メソッド .....	103
6.5.36.	InsertNewPages メソッド .....	103
6.5.37.	MovePages メソッド .....	103
6.5.38.	CopyPages メソッド .....	104
6.5.39.	ResetView メソッド .....	104
6.5.40.	Undo メソッド .....	104
6.5.41.	Redo メソッド .....	104
6.5.42.	Cut メソッド .....	104
6.5.43.	Copy メソッド .....	105
6.5.44.	Paste メソッド .....	105
6.5.45.	Delete メソッド .....	105
6.5.46.	SelectPage メソッド .....	105
6.5.47.	SelectAll メソッド .....	106
6.5.48.	DocumentOpened イベント .....	106
6.5.49.	PageTurned イベント .....	106
6.5.50.	DragDropFiles イベント .....	106
6.5.51.	ContextMenuC イベント .....	106
7.	PDFOutlineCtrl .....	108
7.1.	概要 .....	108
7.2.	アプリケーションが実装すべき項目 .....	108
7.3.	しおりアクションの実行 .....	108
7.4.	API 一覧 .....	109
7.4.1.	プロパティ .....	109
7.4.2.	メソッド .....	109
7.4.3.	イベント .....	109
7.5.	API リファレンス .....	110
7.5.1.	IsOpen プロパティ .....	110
7.5.2.	HasFocus プロパティ .....	110
7.5.3.	DocumentPath プロパティ .....	110
7.5.4.	SDKVersion プロパティ .....	110
7.5.5.	OpenDocument メソッド .....	110

7.5.6.	CloseDocument メソッド .....	111
7.5.7.	GetPDFDocumentHandle メソッド .....	112
7.5.8.	AttachPDFDocument メソッド .....	112
7.5.9.	AttachPDFView メソッド .....	112
7.5.10.	DocumentOpened イベント .....	112
<b>8.</b>	<b>PDFLayerCtrl .....</b>	<b>114</b>
<b>8.1.</b>	<b>概要 .....</b>	<b>114</b>
<b>8.2.</b>	<b>アプリケーションが実装すべき項目 .....</b>	<b>114</b>
<b>8.3.</b>	<b>制限事項 .....</b>	<b>114</b>
<b>8.4.</b>	<b>API 一覧 .....</b>	<b>116</b>
8.4.1.	プロパティ .....	116
8.4.2.	メソッド .....	116
8.4.3.	イベント .....	116
<b>8.5.</b>	<b>API リファレンス .....</b>	<b>117</b>
8.5.1.	IsOpen プロパティ .....	117
8.5.2.	HasFocus プロパティ .....	117
8.5.3.	DocumentPath プロパティ .....	117
8.5.4.	SDKVersion プロパティ .....	117
8.5.5.	OpenDocument メソッド .....	117
8.5.6.	CloseDocument メソッド .....	118
8.5.7.	GetPDFDocumentHandle メソッド .....	119
8.5.8.	AttachPDFDocument メソッド .....	119
8.5.9.	ResetView メソッド .....	119
8.5.10.	DocumentOpened イベント .....	119
<b>9.</b>	<b>改訂履歴 .....</b>	<b>120</b>

## 1. 概要

---

本製品には以下のコントロール（GUI 部品）があります。

- PDFPageViewCtrl (PDF ページ表示・注釈編集)
- PDFThumbnailCtrl (サムネイル表示、ページ編集)
- PDFOutlineCtrl (しおり表示)
- PDFLayerCtrl (レイヤー表示)

開発用の API には以下があります。

- .NET8
- .NET6
- .NET Framework 4.6.2 以降

また、各コントロールには

- 32bit 版
- 64bit 版

があります。

### 1.1. PDFPageViewCtrl

PDFPageViewCtrl の主な機能は以下のとおりです。

カテゴリ	機能項目
PDF ファイル	<ul style="list-style-type: none"><li>● PDF ファイルのオープン・クローズ<ul style="list-style-type: none"><li>➤ PDF1.3～PDF1.7</li><li>➤ PDF2.0 (V5.0)</li></ul></li><li>● ドラッグ &amp; ドロップによるオープン</li><li>● PDF ファイルの保存</li><li>● オープン状態の取得</li><li>● ドキュメントのハンドルを取得</li><li>● ドキュメントのハンドルを別のコントロールへアタッチ</li><li>● PDF Tool API 閲覧制限 PDF の制限表示 (V5.0)</li></ul>
PDF ファイルの情報	<ul style="list-style-type: none"><li>● ドキュメントパスの取得</li><li>● 文書情報の取得</li><li>● ページ数の取得</li></ul>

	<ul style="list-style-type: none"> <li>● ページサイズの取得</li> <li>● ユーザパスワードを持つか、ユーザパスワード取得</li> <li>● オーナパスワードを持つか、オーナパスワードによるロックの解除、オーナパスワードの取得</li> <li>● セキュリティ権限フラグの取得</li> <li>● 文書情報ダイアログの表示</li> <li>● 文書情報、セキュリティ情報の変更</li> </ul>
ページ遷移	<ul style="list-style-type: none"> <li>● 先頭、前、次、最終ページへ移動</li> <li>● マウスホイール、スクロールバー、キー入力、API で任意のページを表示&lt;</li> <li>● 現在のページを取得</li> </ul>
ページ表示	<ul style="list-style-type: none"> <li>● ページレイアウト（連続ページ表示／単ページ表示／見開き表示／表紙を独立表示）</li> <li>● ビューモード（幅に合わせる／ページに合わせる／高さに合わせる／カスタム倍率）</li> <li>● 拡大・縮小 <ul style="list-style-type: none"> <li>➢ より詳細な拡大・縮小倍率（V6.0MR2）</li> </ul> </li> <li>● 縦・横スクロール</li> <li>● ページ表示のアンドゥー、リドゥー</li> <li>● ページ表示の回転</li> <li>● クライアント座標から論理座標への変換（ポイント、矩形（V3.5））</li> <li>● グリッド表示（V6.0）</li> </ul>
選択モード	<ul style="list-style-type: none"> <li>● テキスト選択</li> <li>● ハンドツール</li> <li>● マーキーズーム</li> <li>● カスタムモード</li> </ul>
操作・マウスカーソル	<ul style="list-style-type: none"> <li>● 指定位置へのスクロール</li> <li>● スクロール位置の取得（V6.0）</li> <li>● マウスカーソルの位置取得</li> <li>● コントロールによるマウスカーソル制御の抑止、独自カーソルの設定（V3.5）</li> <li>● 右クリックメニュー、右クリックメニュー表示イベントの取得</li> <li>● マウスイベントの取得</li> <li>● キーボードイベントの取得</li> </ul>

	<ul style="list-style-type: none"> <li>● 自動スクロール (V5.0)</li> <li>● スクロールイベントの取得 (V6.0)</li> </ul>
検索	<ul style="list-style-type: none"> <li>● 検索ダイアログの表示、API による検索</li> <li>● ページ数単位、全文検索</li> <li>● 次や前の検索位置へ移動、検索位置の取得</li> <li>● 検索開始位置の指定 (V3.5)</li> </ul>
印刷	<ul style="list-style-type: none"> <li>● 印刷ダイアログの表示、API による印刷</li> <li>● ページを用紙に合わせる。印刷時の向き、サイズ、印刷位置の調整</li> </ul>
画像/SVG ファイル変換	<ul style="list-style-type: none"> <li>● BMP、PNG、JPEG、TIFF 形式に変換</li> <li>● SVG 形式に変換 (V3.5)</li> <li>● ページ単位、ページ内矩形領域を画像に変換</li> </ul>
テキスト	<ul style="list-style-type: none"> <li>● テキスト選択、コピー</li> <li>● ユーザーによるテキストハイライト、ハイライト色の指定 (V3.5)</li> <li>● 指定範囲のテキスト情報の取得 (V3.5) <ul style="list-style-type: none"> <li>➤ 選択中のテキスト取得 (V5.0)</li> <li>➤ 指定範囲のテキスト選択 (V5.0)</li> <li>➤ 選択中のテキスト情報取得 (V5.0MR1)</li> </ul> </li> </ul>
注釈の作成、編集	<ul style="list-style-type: none"> <li>● ノート注釈、リンク注釈、テキストボックス注釈、タイプライター注釈、ファイル添付、図形・下線・取り消し線など</li> <li>● コピー、カット、ペースト、削除、Ctrl キー&amp;ドラッグによるコピー</li> <li>● 注釈の取得、作成、削除、編集、ヒットテスト</li> <li>● 注釈属性のデフォルト値変更</li> <li>● 注釈外観の保存 (V5.0)</li> <li>● ビュー座標系での位置指定 (V5.0MR1)</li> <li>● 注釈フラグ 編集不可、選択不可 (V5.0MR1)</li> <li>● スタンプ注釈 (標準スタンプ、カスタムスタンプ) (V6.0)</li> <li>● ものさしつール (距離、周辺、面積ツール) (V6.0)</li> <li>● ものさしつール 測定比率と精度ダイアログ (V6.0)</li> <li>● ものさしつールのスナップ機能 (V6.0)</li> <li>● 墨消し (Redact 注釈) (V6.0)</li> <li>● 雲型 (V6.0MR2)</li> </ul>
アンドゥー、リドゥー	<ul style="list-style-type: none"> <li>● アンドゥー、リドゥー</li> </ul>

タッチ操作	<ul style="list-style-type: none"> <li>● スクロール</li> <li>● ピンチイン、ピンチアウト</li> </ul>
イベント	<ul style="list-style-type: none"> <li>● オブジェクトの移動、リサイズ、削除 (V5.0)</li> <li>● テキスト入力の開始／終了、テキストの変更(V5.0)</li> </ul>

## 1.2. PDFThumbnailCtrl

PDFThumbnailCtrl の主な機能は以下のとおりです。

カテゴリ	機能項目
PDF ファイル	<ul style="list-style-type: none"> <li>● PDF ファイルのオープン・クローズ</li> <li>● ドラッグ &amp; ドロップによるオープン</li> <li>● PDF ファイルの保存</li> </ul>
サムネイル表示	<ul style="list-style-type: none"> <li>● サムネイル表示 (複数列表示、注釈表示)</li> <li>● ページ遷移 (マウスホイール、各種キーにデフォルトで対応)</li> </ul>
ページ編集	<ul style="list-style-type: none"> <li>● ページの回転、移動、コピー</li> <li>● ページの削除、カット、コピー&amp;ペースト、移動</li> <li>● 白紙ページの追加</li> <li>● ページサイズの変更</li> <li>● サムネイル上で、ドラッグ &amp; ドロップによるページ移動、コピー</li> <li>● 外部 PDF からページをコピー &amp; ペースト、ドラッグ &amp; ドロップ</li> <li>● アンドゥー、リドゥー</li> </ul>
イベント	<ul style="list-style-type: none"> <li>● 右クリックメニュー、右クリックメニュー表示イベントの取得 (V3.5)</li> </ul>

## 1.3. PDFOutlineCtrl

PDFOutlineCtrl の主な機能は以下のとおりです。 (V5.0 以降)

カテゴリ	機能項目
PDF ファイル	<ul style="list-style-type: none"> <li>● PDF ファイルのオープン・クローズ</li> </ul>
しおり	<ul style="list-style-type: none"> <li>● しおり表示</li> <li>● しおりのカラー・イタリック表示</li> <li>● しおりのアクション実行</li> <li>● 矢印キーによるしおり移動</li> </ul>

## 1.4. PDFLayerCtrl

PDFLayerCtrl の主な機能は以下のとおりです。 (V5.0 以降)

カテゴリ	機能項目
PDF ファイル	<ul style="list-style-type: none"><li>● PDF ファイルのオープン・クローズ</li></ul>
レイヤー	<ul style="list-style-type: none"><li>● レイヤーの階層表示</li><li>● 各レイヤーの表示 ON・OFF</li></ul>

## 2. 動作環境

---

### 2.1. 対応 OS

対応 OS	Windows 11 / 10 64 ビット 日本語版 Windows Server 2025 / 2022 / 2019 日本語版
-------	---

- ※ Microsoft Visual C++ 2019 再頒布可能パッケージが必要です。
- ※ .NET からの利用には .NET Framework 4.6.2 以降 / .NET 6 / .NET 8 のいずれかが必要です。
- ※ Windows Server ではデスクトップエクスペリエンスが必要です。Server Core / Nano Server はサポートされません。
- ※ Microsoft 社のメインストリームサポートが終了したプラットフォームについては、サポート対象外です。

### 2.2. 仮想環境

仮想環境（VMware や Hyper-V など）で実行する場合、実環境上と同じ動作が保証されているのであれば機能制限はありません。

## 3. 制限事項

---

### 3.1. 対象 PDF

- ・ PDF 仕様 (PDF32000-2) に準拠した PDF1.3 ~ PDF2.0
- ・ オープンできる PDF のファイルサイズは 2GB までとなります。
- ・ ページ数は明示的な制限はしておりません。

### 3.2. コントロール全般

- ・ GUI アプリケーションでの使用を前提としています。GUI がないコマンドラインやサービスなどでの動作は保証しておりません。
- ・ リモートデスクトップ接続では注釈の表示が粗くなります。（GDI+の制限事項に依るものです）
- ・ スクロール、ピンチイン・ピンチアウト以外のタッチ操作は動作保証しておりません。また、任意のタッチパネル対応機器での動作は保証しておりません。お使いの機器で動作確認のうえご利用ください。弊社で確認済みの機器は下記です。

- Microsoft Surface Pro 2 / Windows 10 (64bit)

- ・ ActiveX コントロールについては、V6.0 より廃止になりました。
- ・ 注釈の外観情報は他のリーダーと同一にはなりません。注釈は編集可能な注釈のみ対応しています。一部の注釈のプロパティに対応していない場合があります。
- ・ 下記の PDF は開くことは可能ですが、編集については制限がございます。
  - 権限パスワードで保護された PDF は開くことは可能ですが、正しいパスワードを入力してロック解除しないと編集できません。
  - 電子署名された PDF は開くことは可能ですが、編集できません。
  - 閲覧制限された PDF は開くことは可能ですが、編集できません。
  - PDF/A や PDF/X に対して編集を行った場合、PDF/A や PDF/X への準拠は保持されません。
- ・ レイヤーコントロールの制限事項について「8.3 制限事項」を参照ください。
- ・ 標準スタンプを回転したページに配置し不透明度を設定すると、Adobe Acrobat Reader で表示したときに、選択枠の表示がおかしい現象があります。こちらは Adobe Acrobat Reader の問題と思われます。
- ・ ものさしツールの測定値は計算誤差などにより正確にならない場合があります。
- ・ ものさしツールの画面上の表示と PDF の外観が一致しない場合があります。また他のリーダーの表示とは必ずしも一致しません。
- ・ 複雑な PDF の場合、ものさしのスナップ機能を使用するとページ表示に時間がかかる場合があります。
- ・ テキスト抽出時の空白マークや改行マークは正しく挿入されない場合があります。また挿入の仕様については今後変更の可能性があります。
- ・ フォーム XObject に不透明度やブレンドモードが指定されている場合はテキスト検索やテキスト抽出ができません。
- ・ 雲型注釈には以下の制限事項があります。
  - PDF 保存時に外観ストリームは出力しません。PDF リーダーによってはまったく表示されない／正しく表示されない場合があります。
  - 雲型の形状は他の PDF リーダーと同一にはなりません。
  - 雲形注釈の値によっては正しく表示されない場合があります。

### 3.3. レンダリング品質および性能について

- ・ PDF ファイルや実行環境など条件によっては、表示仕様に記載されていない未知の制限が存在する場合があります。
- ・ 表示や画像出力や印刷のレンダリング品質や速度や消費メモリなどの性能について、特定の品質や性能を保証するものではありません。特に Adobe Acrobat/Adobe Reader を含む他社製品の PDF ビューアと同等の品質や性能を保証するものではありません。

- 表示や画像出力や印刷の品質および性能は、実行環境やプリンタおよびプリンタドライバ、PDF ファイルのデータ、描画領域のサイズ、解像度、カラー、拡大縮小率、その他要因によって影響を受けます。条件が違う場合には、必ずしも同じ結果になりません。
- 表示と画像出力、印刷では異なる結果になる場合があります。
- レンダリング品質や性能について、実運用環境と同等の環境や出力方法、また実際に扱うものと同等の PDF データで導入前によくご確認ください。
- 下記の場合に表示や印刷が遅くなります。
  - 半透明やブレンドを含む PDF
  - 細かなパターンやパスを多数含む PDF
  - 注釈オブジェクトの数が多い場合
- 用紙サイズの大きな PDF ファイルや、PDF データ内に大きな画像を含む場合など、レンダリング処理中にメモリが不足した場合、ページ全体もしくはページの一部が描画されないことがあります。
- 32bit 版、とくに.NET Framework、Visual Basic のランタイムを併用する場合はメモリ不足に陥りやすいため、そのような場合は、64bit 版やネイティブアプリケーションをご検討ください。
- タスクスケジューラやサービス等、デスクトップが存在しない環境では、場合によって PDF が正しく印刷や画像出力されない可能性があります。
- PDF/A、PDF/X 等の PDF ファイルの表示は可能ですが、表示に関する要件を満たさない場合があります。特に PDF のプロファイルには厳密に従っていないため、色が正確ではない場合があります。
- モノクロ二値のデバイスに表示・印刷した場合や、画像ファイル出力において、モノクロ (IMAGECOLOR\_MONOCHROME) 、ブラック (IMAGECOLOR\_BLACK) を指定した場合は、十分なレンダリング品質が得られません。
- ピクセル単位での描画の正確性は保証しておりません。特にアンチエイリアスのかかり、線の幅の均一さなどを希望通りとならないことがあります。

### 3.4. 複数プロセス/スレッドでの同時使用

- 複数のプロセスで PDF Viewer Ctrl を同時に使用することは問題ありません。
- 同一プロセス内の複数のスレッドでの PDF Viewer Ctrl の同時使用または同時アクセスには対応しておりません。

## 4. コントロール情報

### 4.1. .NET/.NET Framework 版

.NET 版と.NET Framework 版では利用条件が異なります。情報を下記にまとめました。

	.NET8	.NET6	.NET Framework
ネームスペース	AvsDotNetGuiCtl	AvsDotNetGuiCtl	AvsDotNetGuiCtl
DLL 名	AvsGuiCtl60Net80.dll	AvsGuiCtl60Net60.dll	AvsDotNetGuiCtl.dll
対象フレームワーク	.NET8	.NET6	.NET Framework 4.6.2 以上
検証済み環境	Visual Studio 2022 Visual C# Windows フォームアプリ (.NET8)	Visual Studio 2022 Visual C# Windows フォームアプリ (.NET6)	Visual Studio 2019 Visual C# Windows フォームアプリケーション (.NET Framework)

また、Visual Studio でサポートされる開発作業もバージョンにより異なります。

開発内容	VS2019	VS2022
.NET Framework Windows フォームアプリ		
32bit/64bit アプリ開発	○	○
32bit フォームデザイン／ツールパネルへのコントロール登録	○	×
64bit フォームデザイン／ツールパネルへのコントロール登録	×	○
.NET Windows フォームアプリ		
32bit/64bit アプリ開発	×	○
32bit フォームデザイン／ツールパネルへのコントロール登録	×	×
64bit フォームデザイン／ツールパネルへのコントロール登録	×	×

- ※ マイクロソフト社によりサポートされる環境であれば上記以外でも動作しますが、弊社で動作検証済みの環境は上記となります。
- ※ PDF Viewer Ctrl は Visual Studio 2019、C++/CLI を利用してネイティブライブラリとして開発されており、アーキテクチャ (x86/x64) に合わせたバイナリを使用する必要があります。

- ※ 上位アプリケーションのアーキテクチャは「x86」または「x64」で開発されることを推奨いたします。  
「ANY CPU」は推奨しておりません。
- ※ 開発環境では環境変数”Path”に bin¥Win32 または bin¥x64 フォルダのパスを追加してください。これをおこなわないとビルド時に参照エラーとなったり、VisualStudio のフォームデザイナを開いたときエラーになる場合があります。
- ※ 64bit コントロールの開発には制限がございます。詳しくは「64bit コントロールについて」を参照ください。
- ※ .NET8 および .NET6 の Windows フォームアプリの開発ではフォームデザイナーは利用できません。  
.NET では、フォームデザイナーでネイティブ DLL を利用するコントロールがサポートされないためです。  
このため GUI の開発には直接コードを記述する、もしくは .NET Framework のフォームデザイナーでデザインし生成されたコードを流用するなどの工夫が必要になります。なお、WPF など直接フォームデザイナーにコントロールの DLL をロードしない場合は問題ありません。
- ※ UWP/Windows ストアアプリの開発には対応しておりません。

#### 4.2. 動作に必要なモジュール、手続き等

依存するファイルは bin¥Win32 または bin¥x64 フォルダに含まれる下記ファイルになります。

ご利用になる API に応じて必要なモジュールをご利用ください。

インターフェース	必要なモジュール
.NET8 (32bit)	bin¥Win32¥ AvsGuiCtl60Net80.dll ijwhost.dll 共通モジュール (32bit) .NET デスクトップランタイム 8.0 (x86)
.NET8 (64bit)	bin¥x64¥ AvsGuiCtl60Net80.dll ijwhost.dll 共通モジュール (64bit) .NET デスクトップランタイム 8.0 (x64)
.NET6 (32bit)	bin¥Win32¥ AvsGuiCtl60Net60.dll ijwhost.dll 共通モジュール (32bit) .NET デスクトップランタイム 6.0 (x86)
.NET6 (64bit)	bin¥x64¥ AvsGuiCtl60Net60.dll ijwhost.dll

	共通モジュール (64bit) .NET デスクトップランタイム 6.0 (x64)
.NET Framework (32bit)	bin¥Win32¥ AvsDotNetGuiCtl.dll 共通モジュール (32bit) .NET Framework 4.6.2 以上 (x86)
.NET Framework (64bit)	bin¥x64¥ AvsDotNetGuiCtl.dll 共通モジュール (64bit) .NET Framework 4.6.2 以上 (x64)
共通モジュール (32bit)	bin¥Win32¥ AvsGuci60.dll AvsAHCertificate40.dll AvsAHCommon16.dll AvsAHDMC16.dll AvsAHEditBox11.dll AvsAHFontService17.dll AvsAHGdiPlus11.dll AvsAHGraphicService12.dll AvsAHMFCCCommon12.dll AvsAHPDFEditLib10.dll AvsAHPDFFixUp11.dll AvsAHPDFLib40.dll AvsCommon60.dll AvsEngine60.dll AvsFont60.dll AvsGdiCtl60.dll AvsGraphic60.dll AvsPDFCreator60.dll AvsPDFLinearizer60.dll AvsPdfReader60.dll AvsPDFRes60.dll AvsPdfTk71.dll AvsPdfTkEx71.dll AvsPDFToolPage60.dll AvsRender60.dll AvsSVGCreator60.dll AvsText60.dll AvsXfoCommon60.dll icuin55.dll icuuc55.dll icudt55.dll ljwhost.dll

	<p>base2 (フォルダ) JapanColor2001Coated.icc <a href="#">Visual Studio 2015、2017、2019、および 2022 用 Microsoft Visual C++ 再頒布可能パッケージ(x86)</a></p>
共通モジュール (64bit)	<p>bin¥x64¥ AvsGuiCtl60.dll AvsAHCertificate40.dll AvsAHCommon16.dll AvsAHDMC16.dll AvsAHEditBox11.dll AvsAHFontService17.dll AvsAHGdiPlus11.dll AvsAHGraphicService12.dll AvsAHMFCCCommon12.dll AvsAHPDFEditLib10.dll AvsAHPDFFixUp11.dll AvsAHPDFLib40.dll AvsCommon60.dll AvsEngine60.dll AvsFont60.dll AvsGdiCtl60.dll AvsGraphic60.dll AvsPDFCreator60.dll AvsPDFLinearizer60.dll AvsPdfReader60.dll AvsPDFRes60.dll AvsPdfTk71.dll AvsPdfTkEx71.dll AvsPDFToolPage60.dll AvsRender60.dll AvsSVGCreator60.dll AvsText60.dll AvsXfoCommon60.dll icuin55.dll icuuc55.dll icudt55.dll base2 (フォルダ) JapanColor2001Coated.icc <a href="#">Visual Studio 2015、2017、2019、および 2022 用 Microsoft Visual C++ 再頒布可能パッケージ(x64)</a></p>

#### 4.3. 64bit コントロール

32bit/64bit の取扱い

- PDF Viewer Ctrl は C++で開発されたネイティブのコントロール DLL を内部で利用しているため、32bit・64bit のアーキテクチャに依存します。
- Visual Studio での開発時には以下の場面で 32bit/64bit の取扱いに注意する必要があります。

開発の場面	説明	対処
VS2019 フォームデザイナー上でのデザイン (.NET Framework)	<ul style="list-style-type: none"> <li>Windows フォームアプリケーションのデザイン時には Visual Studio が参照する PATH 環境変数上にコントロールの関連 DLL が存在している必要があります。</li> <li>Visual Studio 2019 は 32bit アプリケーションですので 32bit 版しかフォームデザイナーで利用できません。</li> </ul>	<ul style="list-style-type: none"> <li>Windows フォームデザイン時には、32bitDLL (bin\Win32) を PATH に追加した状態で、Visual Studio を起動します。</li> <li>64bit 版の Windows フォームアプリケーション開発時にコントロールのレイアウトを行う場合は、32bit 版の Windows フォームアプリケーションとして開発・レイアウトを行ったのちに、64bit 版としてビルドしてください。</li> </ul>
VS2022 フォームデザイナー上でのデザイン (.NET Framework)	<ul style="list-style-type: none"> <li>Windows フォームアプリケーションのデザイン時には Visual Studio が参照する PATH 環境変数上にコントロールの関連 DLL が存在している必要があります。</li> <li>Visual Studio 2022 は 64bit アプリケーションですので 64bit 版しかフォームデザイナーで利用できません。</li> </ul>	<ul style="list-style-type: none"> <li>Windows フォームデザイン時には、64bitDLL (bin\x64) を PATH に追加した状態で、Visual Studio を起動します。</li> <li>32bit 版の Windows フォームアプリケーション開発時にコントロールのレイアウトを行う場合は、64bit 版の Windows フォームアプリケーションとして開発・レイアウトを行ったのちに、32bit 版としてビルドしてください。</li> </ul>
Visual Studio 上でのデバッグ実行	<ul style="list-style-type: none"> <li>実行時にプラットフォーム (x86、x64) に合わせた SDK の関連 DLL が参照できる必要があります。</li> </ul>	<ul style="list-style-type: none"> <li>プロジェクトの出力先に、x86、x64 のそれぞれの出力先フォルダに、bin\x86、bin\x64 以下の関連 DLL やリソースをコピーしてください。</li> <li>もしくは、x86、x64 にあわせて、PATH 環境変数に bin\x86、bin\x64 をそれぞれ追加して Visual Studio がそれを参照できるようにしてください。 (x86、x64 にあわせて切り替える。同時に指定されていると動作しません。)</li> </ul>
プロジェクトへの参照追加	プロジェクト上でプラットフォーム (x86、x64) にあわせた DLL のファイル参照を追加する必要があります。	<ul style="list-style-type: none"> <li>Visual Studio 上ではプラットフォーム個別に DLL のファイル参照を指定する事ができません。「プロジェクトへの参照追加について」に記載した方法で</li> </ul>

		ラットフォームにあわせて指定してください。
--	--	-----------------------

#### プロジェクトへの参照追加について

- PDF Viewer SDK の DLL を参照するとき、ビルド対象のプラットフォームが"x86"であるか"x64"であるかにあわせて、それぞれ 32bit 版 DLL (<インストール先>\bin\Win32\AvsDotNetGuiCtl.dll) 、 64bit 版 DLL (<インストール先>\bin\x64\AvsDotNetGuiCtl.dll) を参照する必要があります。
- Visual Studio 2019 の .NET Framework プロジェクト上で、プラットフォームに合わせて別の DLL 参照を指定することはできません。このため、例えば次のような方法で切り替える必要があります。
- 1つめの方法は、.NET Framework プロジェクトを "x86"、"x64" の両方用意し、各プロジェクトでは、それぞれターゲットに応じた DLL を参照します。ソリューションファイルでは、"x86"、"x64" ターゲットにあわせて、追加するプロジェクトを x86 版、x64 版と切り替えるよう構成します。この方法の利点は、プロジェクトファイルを直接編集する必要がなく、Visual Studio 上で編集できることです。
- 2つめの方法は、.NET Framework プロジェクトの DLL 参照を手動で編集する方法です。C#の場合、".csproj" ファイルにおける参照設定を `Include` 属性でターゲットにあわせてそれぞれ指定します。

```
<Reference Include="AvsDotNetGuiCtl">
  <HintPath Condition=" '$(Platform)' == 'x86' ">..\..\Windows-
VS2019\Win32\Release\AvsDotNetGuiCtl.dll</HintPath>
  <HintPath Condition=" '$(Platform)' == 'x64' ">..\..\Windows-
VS2019\x64\Release\AvsDotNetGuiCtl.dll</HintPath>
</Reference>
```

ここでは HintPath で条件指定を行い切り替えていますが、ItemGroup で切り替えることも可能です。

この方法の利点は x86、x64 それぞれプロジェクトを用意する必要がない点です。一方手動で編集するため、間違いややすい、Visual Studio で不意に編集を行うと編集が上書きされてしまう等のデメリットがあります。プロジェクトファイルに詳しくなければ 1 つめの方法をお勧めいたします。SDK 添付のサンプルプロジェクトではこちらの方法を使用しています。

参考) <https://docs.microsoft.com/ja-jp/visualstudio/ide/how-to-configure-projects-to-target-platforms?view=vs-2019>

#### コーディング上の注意点

- DocumentOpen イベントなどで受け取る PDFDocument のハンドルは IntPtr 型としてください。  
64bit の場合は 64bit 整数になるので、Int32 では例外が発生します。

#### 4.4. 座標系について

- PDFViewerCtrl で扱う座標系の種類

以下の 3 つの座標系があります。

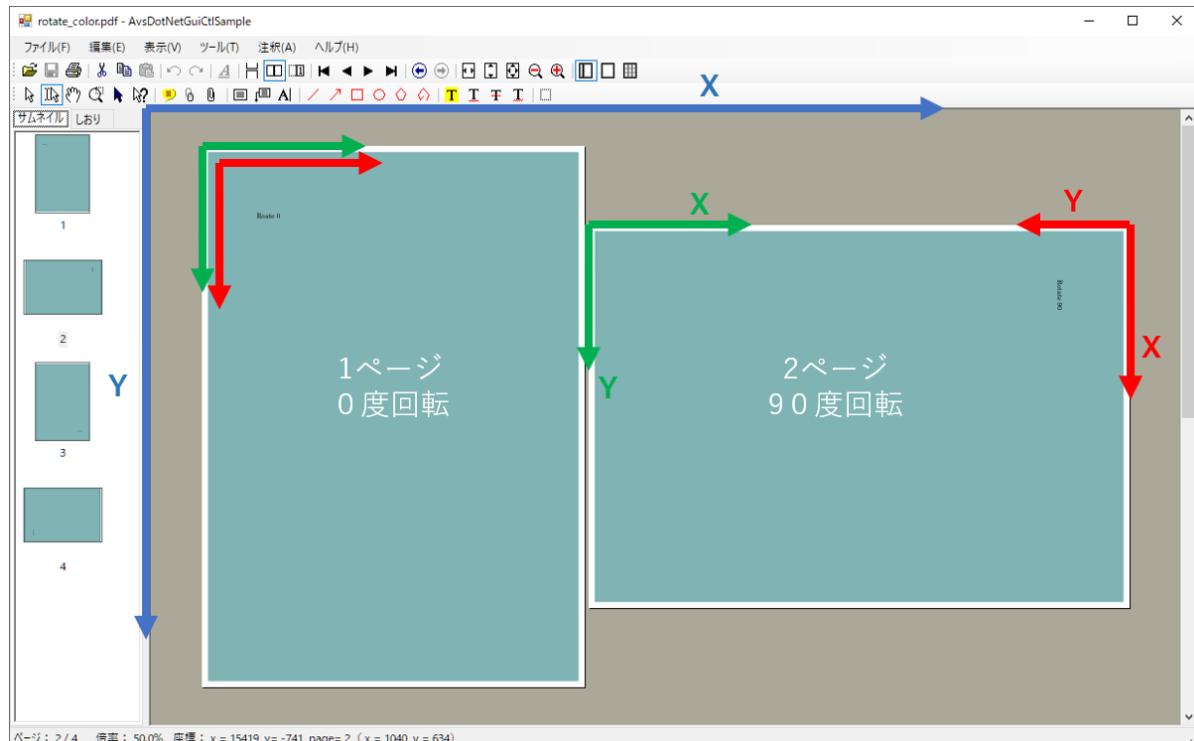
- ・ クライアント座標 … PDFPageViewCtrl のコントロール上の座標
- ・ 論理座標 … PDF ページの座標。ページ回転によって表示上の原点位置は異なる。
- ・ ビュー座標 … PDF ページの表示上の座標。ページ回転によらない。

クライアント座標はコントロール上の座標で X 座標と Y 座標で一意に決まります。一方、論理座標およびビュー座標はページ上の座標です。LogicalPoint クラスで表現され、ページ番号と X 座標と Y 座標を含みます。

	クライアント座標	論理座標	ビュー座標	PDF Viewer API
原点	コントロールの左上	用紙の左上	表示上の用紙左上	表示上の用紙左上
座標軸正方向	表示上の右下	用紙の右下	表示上の右下	表示上の右下
長さの単位	TWIP	TWIP	TWIP	TWIP

※ TWIP 1TWIP = 1/20 ポイント = 1/1440 インチ

連続ページや見開き表示など、コントロールの表示状態によって、クライアント座標と、論理座標およびビュー座標との対応関係は実行時に変化します。下記の図において、青はクライアント座標系、赤は論理座標系、緑はビュー座標系を表しています。



- ・ クライアント座標  
HitTest メソッドや MouseDownC イベント等で取得される位置はクライアント座標であるものがあります。  
クライアント座標から論理座標／ビュー座標に変換するには ClientToLogical メソッドを利用してください。
- ・ 論理座標

論理座標は用紙の左上原点(0, 0)で右下方向にプラス、単位は TWIP (1TWIP = 1/20 ポイント) になります。表示上の左上が原点となる PDF Viewer API の座標系とは若干異なることになりますので注意してください。

- ・ ビュー座標

ビュー座標系は表示上の原点が用紙の左上となります。表示上の位置指定を行う場合はこちらを利用すると便利です。

- ・ 論理座標とビュー座標の切り替え

論理座標とビュー座標は実行に ViewCoord プロパティで切り替えることができます (false…論理座標、true …ビュー座標)。注釈の作成など位置座標を指定する場合に、アプリケーションに都合のよい座標系をご利用ください。コントロール内部で指定された座標系にあわせて相互変換を行います。

■ViewCoord プロパティの影響を受けるプロパティ、メソッド

名称	影響を受けるデータ
ClientToLogical	戻り値 (LogicalPoint)
ClientToLogicalRect	戻り値 (System.Drawing.Rectangle)
GetCursorPosition	戻り値 (LogicalPoint)
ScrollToPosition	引数 IPos (LogicalPoint)
CreateObject	引数 object (AvsObject) の Coordinates プロパティ
ModifyObjects	引数 object (AvsObject) の Coordinates プロパティ ※ids 引数で指定される注釈オブジェクトが存在するページが異なる場合、最初の注釈オブジェクトのページ回転にあわせて座標を解釈します。 注釈オブジェクトの Coordinates プロパティの編集時に注釈が存在するページが異なる場合は、複数回の呼び出しに分けて指定してください。
GetObjectInfo	戻り値(AvsObject) の Coordinates プロパティ
GetTextInfoInRect	戻り値 (ArrayList) の要素 AvsTextInfo の Rect プロパティ、WritingMode プロパティ
GetSelectedTextInfo	戻り値 (ArrayList) の要素 AvsTextInfo の Rect プロパティ、WritingMode プロパティ
SelectTextString	引数 start (LogicalPoint) 、 end (LogicalPoint)
MakeBmpPage	引数 rcClip (System.Drawing.Rectangle)
MakeJpegPage	引数 rcClip (System.Drawing.Rectangle)
MakePngPage	引数 rcClip (System.Drawing.Rectangle)
MakeSvgPage	引数 rcClip (System.Drawing.Rectangle)
FitViewToRect	引数 IPos1, IPos2 (LogicalPoint)

※GetPageWidth/GetPageHeight メソッドは常に論理座標系でのページ幅、高さを返します。

※MakeTiffPage メソッドは常に論理座標系にて rcClip 引数を解釈します。

#### 4.5. タッチ操作

- ・ タッチ操作による上下左右のスクロール、PDF 表示画面のピンチイン、ピンチアウトが可能です。
- ・ タッチ操作が可能なモードはテキスト選択モード (ASM\_TXTSEL) です。
- ・ テキスト選択モードでテキストの選択が邪魔になる場合は、下記の対応が可能です。
  - 注釈編集が不要な場合は、CtrlStyle プロパティに AGCS\_PREVIEWMODE を指定しプレビューモードにします。プレビューモードではテキスト選択が無効になります。
  - 注釈編集が必要な場合、AEO\_PANSCTRL を指定したカスタムモードを **SetSelectModeEx** メソッドで作成し、アプリケーションでモードを適宜切り替えます。
- ・ 以下の制限があります。
  - スクロール、ピンチイン、ピンチアウト以外の操作は動作保証しておりません。
  - タッチ操作による右クリックは対応しておりません。
  - ペンによる操作は動作保証しておりません。
  - 任意のタッチパネル対応機器での動作は保証しておりません。弊社の動作確認環境について「対応 OS」—「制限事項」を参照ください。

#### 4.6. PDF 保存時の暗号化

- ・ ユーザーパスワードまたはオーナーパスワードが指定されていると、PDF 保存時に PDF は暗号化されます。暗号方式は PDF バージョンによって下記となります。

PDF バージョン	暗号方式
1.3	40bit RC4 ※
1.4	128bit RC4
1.5	128bit RC4
1.6	128bit AES
1.7	256bit AES
2.0	256bit AES

※ユーザーパスワードのみの場合、128bit RC4 となります。

#### 4.7. 文字列検索

- ・ MediaBox や CropBox の指定に関わらず Box 外の文字列も検索にヒットします。

#### 4.8. オブジェクト ID

- PDFPageViewCtrl では注釈オブジェクトを一意に特定するため、オブジェクト ID を割り当てます。オブジェクト ID は GUID を表す文字列です。
- PDF に注釈が存在し注釈の NM キーが存在する場合はこれをオブジェクト ID とします。NM キーが空の場合や、新規に注釈オブジェクトを作成した場合は GUID を生成して割り当てます。
- PDF 保存時には注釈の NM キーにオブジェクト ID を保存します（V5.0MR1 以降）。オブジェクト ID を NM キーとして保存するのは PDF Viewer Ctrl の仕様です。

##### 注意事項

- PDF 仕様上、注釈の NM キーはオプションデータです。本製品以外の他のソフトウェアで PDF ファイルを編集した場合、オブジェクト ID は保証されない場合があります。
- PDF ファイルにおいて NM キーが重複するなど一意ではない場合、誤動作の可能性があります。
- ポップアップ注釈のオブジェクト ID は読み込み時に毎回新しい値が設定され、一意になりません。
- PDF Viewer Ctrl が対応していない注釈、編集対象であっても保存までに編集が行われなかった注釈のオブジェクト ID は保存されません。

#### 4.9. ファイルオープン時の一時ファイルへのコピー

V6 以降では、オープンしたファイルを一時フォルダにコピーしてから編集するようになりました。

作成される一時ファイルは以下のようになります。

一時ファイルの種類	ファイル名	V5	V6
オープン時のコピー	%TEMP%¥ava_~.tmp	なし	あり
保存時のコピー	%TEMP%¥tmpXXXX.tmp.pdf	あり	あり

※ファイル名は予告なく変更される可能性があります。

詳しくは、以下の動作になります。

	V5	V6 以降
ファイルオープン	<ul style="list-style-type: none"><li>指定されたファイルを直接オープンします。</li><li>ファイルはコピーされません。</li></ul>	<ul style="list-style-type: none"><li>指定された PDF を一時フォルダにコピーします。</li><li>コピーした一時ファイルをオープンします。</li><li>ファイル名は%TEMP%¥avs_~.tmp の形式になります。（現状の仕様。変更の</li></ul>

		可能性あり)
DocumentPath	指定した入力ファイルパス	指定した入力ファイルパス
TemporaryPath	API なし	一時ファイルのファイルパス
上書き保存	DocumentPath に上書き	DocumentPath に上書き (一時ファイルは更新されない)
保存中の一時ファイル	作成する	作成する
ファイルクローズ	—	クローズ後、TemporaryPath を削除します。
ファイルのロック	オープン中 DocumentPath のファイルがロックされます。	オープン中 TemporaryPath のファイルがロックされます。 保存中、DocumentPath のファイルがロックされます。

## 5. PDFPageViewCtrl

---

### 5.1. 概要

PDF のページを表示するコントロールです。様々なページレイアウト、ビューモード、選択ツール、検索、回転、印刷、画像ファイル変換といった機能を利用することができます。

フォームに貼り付けるだけで簡単に利用することができます。

### 5.2. チュートリアル .NET Framework 版

以下では VisualStudio C#での初步的な使用方法を説明します。.NET Framework 版を利用するには事前に.NET Framework 4.6.2 以降のインストールが必要です。 詳細は「4.2 動作に必要なモジュール、手続き等」をご覧ください。以下は 32bit 版での説明です。

- ① VisualStudio を起動する前に環境変数"Path"に bin¥Win32 フォルダのパスを追加してください。こうしないとフォームデザイナを開いたときエラーになることがあります。（ビルド、実行には影響しませんので開発環境のみおこなってください。64bit 版コントロールでの開発については、「64bit 版コントロールについて」を参照ください。）
- ② メニューから[ファイル]-[新規作成]-[プロジェクト]を選び、「新しいプロジェクト」ダイアログから VisualC# の「Windows フォーム アプリケーション」を選択して適当なプロジェクト名（ここでは"AvsDotNetGuiSample"とします）を入力したら[OK]を押してください。
- ③ メニューから[ツール]-[ツールボックス アイテムの選択]を選び、右下の「参照...」ボタンを押して bin¥Win32 ディレクトリにある AvsDotNetGuiCtl.dll を選択してください。「PDFPageViewControl」にチェックをつけて[OK]を押します。
- ④ ツールボックススペインに「PDFPageViewControl」のアイコンが追加されていますのでそれを選びフォーム上でドラッグ & ドロップして貼り付けてください。その他、 OpenFileDialog コントロール、Button コントロールを貼り付けてください。
- ⑤ 貼り付けた Button コントロールをダブルクリックします。コードが表示されますので次のように処理を書きます。

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
        pdfPageViewCtrl1.OpenDocument(openFileDialog1.FileName, "");
}
```

- ⑥ メニューから[プロジェクト]-[参照の追加]を選び[参照]タブを選択し、bin¥Win32 ディレクトリにある AvsDotNetGuiCtl.dll を選択してください。[OK]を押すとプロジェクトの参照設定に AvsDotNetGuiCtl が追加されます。

- ⑦ 実行します。ボタンを押すとファイルダイアログが表示され PDF ファイルを指定すると表示することができます。

### 5.3. アプリケーションが実装すべき項目

コントロールには多くの機能がデフォルトで実装されており、ほとんどの機能はボタンを用意してメソッドを呼び出すだけで、もしくはファイルを開いただけで、簡単に利用できるようになっています。ここでは各機能についてアプリケーション側が実装すべきこと、関知できることについて簡単に説明します。詳しくは付属のサンプルプログラムを参考にしてください。

機能概要	説明
ファイルのドラッグ & ドロップ	デフォルトではドロップされたファイルの一つを開きます。 異なる動作をおこないたい場合は DragDropFiles イベントをハンドリングしてください。
コントロール間の同期（編集）	デフォルトではなにもしません。 同期をとるには DocumentOpened イベントで AttachPDFDocument してください。同期をとるとページの入れ替えや回転といった編集処理をおこなったとき、同期中の全コントロールに反映されます。
コントロール間の同期（ページ遷移）	デフォルトではなにもしません。 同期をとるには PageTurned イベントで GotoPage してください。
右クリックメニュー	デフォルトで用意されています。 ContextMenuC イベントをハンドリングすることで独自の右クリックメニューを使用することも可能です。
ドキュメント情報の編集（セキュリティ解除）	ドキュメント情報編集ダイアログが用意されています。 ShowDocumentPropertyDialog を呼び出してください。独自のダイアログを作成して使用することも可能ですし、ModifyObjects で直接編集も可能です。
注釈の処理	注釈を処理するための GUI インタフェースはデフォルトで用意されています。 プログラムから注釈を操作したい場合は、 CreateObject/GetObjectInfo/DeleteObject/ModifyObjects などを使用してください。SetDefaultObjectProperty でデフォルトプロパティを変更することも可能です。 ショートカットキー[Delete]/[BackSpace]/[Esc]/[Ctrl]/[Shift]/方向キーがデフォルトでハンドリングされています。意図的な別処理への

	置き換えでなければアプリケーション側ではハンドリングしないでください。
アンドゥー、リドゥー	<p>デフォルトで実装されています。</p> <p>独自のアンドゥー処理を差し込むことはできません。</p> <p>ショートカットキー[Ctrl]+[Z]/[Y]がデフォルトでハンドリングされていますので意図的に別の処理に置き換える場合はアプリケーション側ではハンドリングしないでください。</p>
スクロール	<p>スクロールバークリック、ハンドツール、キー押下などによるスクロール処理がデフォルトで実装されています。ScrollToPosition で任意の位置へスクロールすることができます。</p> <p>ショートカットキー[Home]/[End]/[PageUp]/[PageDown]/方向キーがデフォルトでハンドリングされています。意図的な別処理への置き換えでなければアプリケーション側ではハンドリングしないでください。</p>
検索	<p>デフォルトで実装されています。</p> <p>検索ダイアログは用意(ShowSearchDialog)されておりますが、独自のダイアログを作成して使用することも可能です。</p>
カット、コピー&ペースト、すべて選択	<p>デフォルトで実装されています。</p> <p>ショートカットキー[Ctrl]+[X]/[C]/[V]/[A]がデフォルトでハンドリングされています。意図的な別処理への置き換えでなければアプリケーション側ではハンドリングしないでください。</p>
印刷	<p>デフォルトで実装されています。</p> <p>印刷ダイアログは用意されておりますが、独自のダイアログを作成して使用することも可能です。</p>
アクション実行	<p>デフォルトで実装されています。</p> <p>DoAction イベントを実装することでアクション実行時のメッセージを変更したり非表示にすることが可能です。</p>

## 5.4. API 一覧

### 5.4.1. プロパティ

プロパティ名	.NET/.NET Framework	概要
IsOpen	○	PDF がオープンされているか調べます。
IsModified	○	最後に保存してから PDF が変更されているかどうかを調べます。
IsLockedByOwnerPassword	○	オーナーパスワードでロックされている状態かどうか調べます。
IsSearched	○	検索状態か調べます。
IsSearchedAll	○	全検索状態か調べます。
HasFocus	○	コントロールがフォーカスを持っているかどうか調べます。
PageCount	○	ページ数を取得します。
ShowCoverPageDuringTwoUp	○	見開きレイアウトの時、1ページ目を右側に表示します。
PageShadow	○	ページに影をつけます。
BackgroundColor	○	背景色を設定します。
TextSelectedColor	○	テキスト選択色を設定します。
SearchedAllColor	○	全検索で見つかったキーワードの背景色を設定します。
HiligtTextColor	○	テキストのハイライト色を設定します。
PageLayout	○	ページレイアウトを設定します。
ViewMode	○	ビューモードを設定します。
SelectMode	○	選択モードを設定します。
Direction	○	見開きレイアウトの綴じ方を取得/設定します。
CtrlStyle	○	スタイルを設定します。
HasUserPassword	○	ユーザーパスワードが設定されているかどうかを調べます。
HasOwnerPassword	○	オーナーパスワードが設定されているかどうかを調べます。
UserPassword	○	ユーザーパスワードを取得/設定します。
OwnerPassword	○	オーナーパスワードを取得/設定します。
SecurityPermissonFlag	○	PDF のセキュリティフラグを取得/設定します。

ZoomRatio	<input type="radio"/>	ズーム倍率を設定します。
ZoomRatioLimit	<input type="radio"/>	最大ズーム倍率を取得します。
CurrentPageNumber	<input type="radio"/>	現在のページ番号を表示します。
CanUndo	<input type="radio"/>	Undo が可能か調べます。
CanRedo	<input type="radio"/>	Redo が可能か調べます。
CanCut	<input type="radio"/>	Cut が可能な状態か調べます。
CanCopy	<input type="radio"/>	Copy が可能な状態か調べます。
CanPaste	<input type="radio"/>	Paste が可能な状態か調べます。
CanDelete	<input type="radio"/>	Delete が可能な状態か調べます。
CanModifyPage	<input type="radio"/>	ページ編集が可能な状態か調べます。
CanModifyAnnotation	<input type="radio"/>	注釈の追加、削除、変更が可能な状態か調べます。
CanModifyFontStyle	<input type="radio"/>	フォントスタイル変更が可能な状態か調べます。
CanUndoView	<input type="radio"/>	UndoView が可能な状態か調べます。
CanRedoView	<input type="radio"/>	RedoView が可能な状態か調べます。
CanPrint	<input type="radio"/>	印刷が可能な状態か調べます。
SelectedObjectCount	<input type="radio"/>	選択されているオブジェクトの数を取得します。
PdfVersion	<input type="radio"/>	PDF のバージョンを取得します。
CreationDate	<input type="radio"/>	PDF の作成日時を取得します。
ModifyDate	<input type="radio"/>	PDF の更新日時を取得します。
Title	<input type="radio"/>	PDF のタイトルを取得/設定します。
Author	<input type="radio"/>	PDF の作成者を取得/設定します。
Subject	<input type="radio"/>	PDF のサブタイトルを取得/設定します。
Keywords	<input type="radio"/>	PDF のキーワードを取得/設定します。
Creator	<input type="radio"/>	PDF を作成したアプリケーション名を取得します。
Producer	<input type="radio"/>	PDF 変換をおこなったツール名を取得します。
DocumentPath	<input type="radio"/>	オープンしている PDF の Path を取得します。
SDKVersion	<input type="radio"/>	SDK バージョン番号を取得します。
SearchStartPage	<input type="radio"/>	検索開始ページを取得/設定します。
LoadPageCount	<input type="radio"/>	読み込みページ数を設定します。
SelectedTextString	<input type="radio"/>	選択中のテキストを取得します。
ViewCoord	<input type="radio"/>	View 座標であるかを取得/設定します。

IncludeArtifact	<input type="radio"/>	テキスト装飾を含めるかを取得/設定します。
SnapPointType	<input type="radio"/>	ものさしでスナップする点の種類を取得/設定します。
HasRedactAnnotation	<input type="radio"/>	墨消し (Redact 注釈) を含むかを取得します。
ScrollPosition	<input type="radio"/>	スクロール位置を論理座標で取得します。
DeviceScrollPosition	<input type="radio"/>	スクロール位置をクライアント座標で取得します。

#### 5.4.2. メソッド

---

メソッド名	.NET/.NET Framework	概要
OpenDocument	<input type="radio"/>	PDF をオープンします。
SaveDocument	<input type="radio"/>	PDF を保存します。
CloseDocument	<input type="radio"/>	PDF をクローズします。
PrintDocument	<input type="radio"/>	印刷をおこないます。
GetPDFDocumentHandle	<input type="radio"/>	PDFDocument のハンドルを取得します。
AttachPDFDocument	<input type="radio"/>	PDFDocument のハンドルをコントロールに関連付けます。
UnlockOwnerPasswordSecurity	<input type="radio"/>	オーナーパスワードを指定してセキュリティを解除します。
GetPageWidth	<input type="radio"/>	ページの幅を取得します。
GetPageHeight	<input type="radio"/>	ページの高さを取得します。
GetNextZoomRatio	<input type="radio"/>	ズーム倍率の次の値を取得します。
GotoPage	<input type="radio"/>	指定されたページを表示します。
GotoFirstPage	<input type="radio"/>	先頭ページを表示します。
GotoLastPage	<input type="radio"/>	最終ページを表示します。
GotoPreviousPage	<input type="radio"/>	前ページを表示します。
GotoNextPage	<input type="radio"/>	次ページを表示します。
GetPageRotation	<input type="radio"/>	ページの回転角度を取得します。
RotatePages	<input type="radio"/>	ページを回転します。
ResetView	<input type="radio"/>	ビューを作り直します。
Undo	<input type="radio"/>	最後の編集操作を取り消します。
Redo	<input type="radio"/>	Undo を取り消します。

Cut	<input type="radio"/>	切り取ってクリップボードに保存します。
Copy	<input type="radio"/>	コピーしてクリップボードに保存します。
Paste	<input type="radio"/>	クリップボードの内容を貼り付けます。
Delete	<input type="radio"/>	削除をおこないます。
SelectAll	<input type="radio"/>	ページテキストの全選択をおこないます。
ShowDocumentPropertyDialog	<input type="radio"/>	ドキュメントプロパティダイアログを表示します。
ShowFontStyleDialog	<input type="radio"/>	フォントスタイルダイアログを表示します。
ShowObjectPropertyDialog	<input type="radio"/>	オブジェクトプロパティダイアログを表示します。
UndoView	<input type="radio"/>	前の表示に戻します。
RedoView	<input type="radio"/>	UndoView を取り消します。
ShowSearchDialog	<input type="radio"/>	検索ダイアログを表示します。
Search	<input type="radio"/>	検索をおこないます。
SearchNext	<input type="radio"/>	次検索をおこないます。
SearchPrevious	<input type="radio"/>	前検索をおこないます。
ClearSelect	<input type="radio"/>	選択中のテキストをクリアします。
MakeBmpPage	<input type="radio"/>	指定したページを BMP 画像に変換します。
MakeJpegPage	<input type="radio"/>	指定したページを JPEG 画像に変換します。
MakePngPage	<input type="radio"/>	指定したページを PNG 画像に変換します。
MakeTiffPage	<input type="radio"/>	指定したページを TIFF 画像に変換します。
MakeEmfPage	<input type="radio"/>	指定したページを EMF 画像に変換します。
MakeSvgPage	<input type="radio"/>	指定したページを SVG 形式に変換します。
GetDefaultObjectProperty	<input type="radio"/>	デフォルトのオブジェクトプロパティを取得します。
SetDefaultObjectProperty	<input type="radio"/>	デフォルトのオブジェクトプロパティを設定します。
GetCursorPosition	<input type="radio"/>	現在カーソル位置の座標を取得します。
ScrollToPosition	<input type="radio"/>	論理座標で指定した位置へスクロールします。
ScrollToDevicePosition	<input type="radio"/>	クライアント座標で指定した位置へスクロールします。
ClientToLogical	<input type="radio"/>	クライアント座標を論理座標に変換します。
SetSelectionModeEx	<input type="radio"/>	カスタム選択モードを作成するときに使用します。
CreateObject	<input type="radio"/>	オブジェクト（注釈）を作成します。
GetObjectCount	<input type="radio"/>	指定したページにあるオブジェクト（注釈）数を取得します。

GetObjectInfo	<input type="radio"/>	オブジェクトを取得します。
DeleteObjects	<input type="radio"/>	オブジェクトを削除します。
ModifyObjects	<input type="radio"/>	オブジェクトを編集します。
HitTest	<input type="radio"/>	指定した位置になにがあるか調べます。
GetNextObjectId	<input type="radio"/>	次のオブジェクトの ID を取得します。
SetHilightText	<input type="radio"/>	指定範囲のテキストをハイライト表示します。
ClearHilightText	<input type="radio"/>	ハイライト表示をすべて解除します。
GetTextInfoInRect	<input type="radio"/>	指定範囲のテキスト情報を取得します。
ClientToLogicalRect	<input type="radio"/>	矩形をクライアント座標から論理座標に変換します。
SetCtrlCursor	<input type="radio"/>	コントロールのカーソルを指定します。
SetSmoothing	<input type="radio"/>	スムージングを設定します。
SelectTextString	<input type="radio"/>	指定した範囲の文字列を選択します。
SetFocus	<input type="radio"/>	コントロールにフォーカスを設定します。
GetSelectedTextInfo	<input type="radio"/>	選択中のテキストのテキスト情報を取得します。
ShowMeasureDialog	<input type="radio"/>	測定比率と精度ダイアログを表示します。
RedactAndSaveDocument	<input type="radio"/>	墨消しを反映して PDF ファイルを保存します。

#### 5.4.3. イベント

イベント名	.NET /.NET Frame work	概要
DocumentOpened	<input type="radio"/>	ドキュメントがオープンされたときに呼ばれます。
PageTurned	<input type="radio"/>	ページ移動が起きたときに呼ばれます。
Zoom	<input type="radio"/>	拡大・縮小が起きたときに呼ばれます。
DragDropFiles	<input type="radio"/>	ファイルがドラッグ & ドロップされたときに呼ばれます。
ObjectAdded	<input type="radio"/>	オブジェクトが追加された時に呼ばれます。
DoAction	<input type="radio"/>	アクション実行時に呼ばれます。
DefaultObjectPropertyChanged	<input type="radio"/>	デフォルトのオブジェクトプロパティが変更されたときに呼ばれます。

MouseDownC	<input type="radio"/>	コントロール上でマウスボタンが押されたときに呼ばれます。
MouseUpC	<input type="radio"/>	コントロール上でマウスボタンが離されたときに呼ばれます。
DoubleClickC	<input type="radio"/>	コントロール上でマウスボタンをダブルクリックしたときに呼ばれます。
MouseMoveC	<input type="radio"/>	コントロール上でマウスを動かしたときに呼ばれます。
ContextMenuC	<input type="radio"/>	コントロール上でマウスボタンが右クリックされたときやコンテキストメニューボタンが押されたときに呼ばれます。
KeyDownC	<input type="radio"/>	コントロール上でキーが押されたときに呼ばれます。
KeyUpC	<input type="radio"/>	コントロール上でキーが離されたときに呼ばれます。
ObjectDeleted	<input type="radio"/>	注釈が削除されたときに呼ばれます。
ObjectResized	<input type="radio"/>	注釈をリサイズや端点を移動したときに呼ばれます。
ObjectMoved	<input type="radio"/>	注釈を移動したときに呼ばれます。
EditBoxShowed	<input type="radio"/>	テキスト入力ボックスが表示や非表示されたときに呼ばれます。
EditBoxTextEdited	<input type="radio"/>	テキスト入力ボックスに文字が入力されたときに呼ばれます。
Scrolled	<input type="radio"/>	スクロールバーがスクロールされたときに呼ばれます。

## 5.5. API リファレンス

### 5.5.1. IsOpen プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： PDF がオープンされているかどうか調べます。オープンされていれば True になります。

### 5.5.2. IsModified プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： 最後に保存してから PDF が変更されているかどうかを調べます。変更されていれば True になります。

### 5.5.3. IsLockedByOwnerPassword プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： PDF にオーナーパスワードセキュリティによるロックがかかっている状態かどうかを調べます。ロックされていれば True になります。

### 5.5.4. IsSearched プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： 検索状態かどうかを調べます。検索状態であれば True になります。

### 5.5.5. IsSearchedAll プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： 全検索状態かどうかを調べます。全検索状態であれば True になります。

## 5.5.6. HasFocus プロパティ

---

型: bool (.NET) (getのみ)

解説: コントロールがフォーカスを持っているかどうか調べます。フォーカスをもっていれば True になります。.NET の Control.Focused プロパティは正しく機能しませんのでこのプロパティを使用してください。

## 5.5.7. PageCount プロパティ

---

型: int32 (.NET) (getのみ)

解説: PDF の総ページ数です。PDF がオープンされていない場合は 0 になります。

## 5.5.8. ShowCoverPageDuringTwoUp プロパティ

---

型: bool (.NET) (get / set)

解説: 見開きページレイアウトで1ページ目を独立して表示するかどうかの指定です。

デフォルトは False です。

## 5.5.9. PageShadow プロパティ

---

型: bool (.NET) (get / set)

解説: ページの影を描画するかどうかの指定です。描画する場合は True を指定します。デフォルトは True です。

## 5.5.10. BackgroundColor プロパティ

---

型: System.Drawing.Color (.NET) (get / set)

解説: 背景色 (ページの外の領域) です。

### 5.5.11. TextSelectedColor プロパティ

---

型： System.Drawing.Color ([.NET](#)) ([get / set](#))

解説：テキストの選択色です。

### 5.5.12. SearchedAllColor プロパティ

---

型： System.Drawing.Color ([.NET](#)) ([get / set](#))

解説：全検索で見つかったキーワードの背景色です。

### 5.5.13. HilightTextColor プロパティ

---

型： System.Drawing.Color ([.NET](#)) ([get / set](#))

解説：SetHilightText メソッドで指定したテキストハイライトの背景色です。

### 5.5.14. PageLayout プロパティ

---

型： AvsPageLayout (enum) ([get / set](#))

設定値：

AvsPageLayout (enum 値)	値	説明
APL_SINGLE	0	単ページ表示
APL_CONTINUOUS	1	連続ページ表示
APL_TWOUP	2	見開き表示
APL_TWOUP_CONTINUOUS	3	連続見開き表示
APL_TWOUP_COVER	4	見開き表示（表紙）
APL_TWOUP_COVER_CONTINUOUS	5	連続見開き表示（表紙）

解説：ページレイアウトです。 連続表示（1と3）はビューモードの「ページサイズに合わせる」と排他の関係にあります。 よって連続表示を指定すると「ページサイズに合わせる」ビューモードは「指定倍率」に変更されます。

デフォルトは「連続ページ表示」です。

### 5.5.15. ViewMode プロパティ

---

型 : AvsViewMode (enum) [\(get / set\)](#)

設定値 :

AvsViewMode	値	説明
AVM_CUSTOM	0	指定倍率
AVM_FITPAGE	1	ページサイズに合わせる
AVM_FITWIDTH	2	幅に合わせる
AVM_FITHEIGHT	3	高さに合わせる

解説 : ビューモードです。 「ページサイズに合わせる」はページレイアウトの連続表示と排他の関係になります。

よって「ページサイズに合わせる」を指定すると連続表示ページレイアウトは単表示に変更されます。

デフォルトは「幅に合わせる」です。

### 5.5.16. SelectMode プロパティ

---

型 : AvsSelectMode (enum) [\(get / set\)](#)

設定値 :

AvsSelectMode	値	説明
ASM_NORMAL	0	通常モード オブジェクトの矩形選択が可能です。
ASM_TXTSEL	1	テキスト選択モード  ドラッグすることによってテキストを選択することができます。  ALT を押しながらドラッグすると矩形選択が可能で す。
ASM_HAND	2	ハンドモード  ビューをドラッグすることによってスクロールする ことができます。
ASM_MQZOOM	3	マーキーズームモード

		ページの一部分をドラッグで範囲指定するとその範囲がビュー全体に表示されるように拡大されます。また、単にクリックするとクリックした位置を中心拡大します。 Ctrl を押した状態でおこなうと縮小になります。
ASM_OBJEDIT	4	オブジェクト編集モード リンク注釈を編集することができます。オブジェクトの矩形選択が可能です。
ASM_CUSTOM1 - ASM_CUSTOM5	51-55	カスタムモード 独自の選択モードを作成することができます。
ASM_INS_STICKYNODE	100	ノート注釈を挿入するモードです。
ASM_INS_LINK	101	リンク挿入するモードです。
ASM_INS_FILEATTACHMENT	102	ファイル添付注釈を挿入するモードです。
ASM_INS_TEXTBOX	103	テキストボックス注釈を挿入するモードです。
ASM_INS_CALLOUT	104	引出線注釈を挿入するモードです。
ASM_INS_TYPEWRITER	105	タイプライター注釈を挿入するモードです。
ASM_INS_LINE	106	線注釈を挿入するモードです。
ASM_INS_ARROW	107	矢印線注釈を挿入するモードです。
ASM_INS_RECTANGLE	108	長方形注釈を挿入するモードです。
ASM_INS_OVAL	109	橢円注釈を挿入するモードです。
ASM_INS_POLYGON	110	多角形注釈を挿入するモードです。
ASM_INS_POLYLINE	111	折れ線注釈を挿入するモードです。
ASM_INS_HIGHLIGHT	112	ハイライト注釈を挿入するモードです。
ASM_INS_UNDERLINE	113	下線注釈を挿入するモードです。
ASM_INS_CROSSOUT	114	取消線注釈を挿入するモードです。
ASM_INS_SQUIGGLY	115	波線注釈を挿入するモードです。
ASM_INS_STAMP_APPROVED	116	スタンプ注釈（承認済）を挿入するモードです。
ASM_INS_STAMP_DONE	117	スタンプ注釈（完了）を挿入するモードです。
ASM_INS_STAMP_TOPSECRET	118	スタンプ注釈（極秘）を挿入するモードです。
ASM_INS_STAMP_DRAFT	119	スタンプ注釈（草稿）を挿入するモードです。
ASM_INS_STAMP_FINAL	120	スタンプ注釈（最終）を挿入するモードです。
ASM_INS_STAMP_EVALUATION	121	スタンプ注釈（評価用）を挿入するモードです。

ASM_INS_STAMP_PUBLIC	122	スタンプ注釈（公開用）を挿入するモードです。
ASM_INS_STAMP_REFERENCE	123	スタンプ注釈（参照用）を挿入するモードです。
ASM_INS_STAMP_DISMISSAL	124	スタンプ注釈（却下）を挿入するモードです。
ASM_INS_STAMP_PRIVATE	125	スタンプ注釈（非公開）を挿入するモードです。
ASM_INS_STAMP_PRELIMINARY	126	スタンプ注釈（試用結果）を挿入するモードです。
ASM_INS_STAMP_VOID	127	スタンプ注釈（無効）を挿入するモードです。
ASM_INS_LINE_DIMENSION	135	距離ツールを挿入するモードです。
ASM_INS_POLYGON_DIMENSION	136	周辺ツールを挿入するモードです。
ASM_INS_POLYLINE_DIMENSION	137	面積ツールを挿入するモードです。
ASM_INS_STAMP_CUSTOM	138	スタンプ注釈（カスタム）を挿入するモードです。
ASM_INS_REDACT	139	Redact 注釈を挿入するモードです。
ASM_INS_CLOUD	140	雲型注釈（多角形）を挿入するモードです。

解説：選択モードです。デフォルトは「テキスト選択モード」です。

#### 5.5.17. Direction プロパティ

---

型 : AvsDirectionType (enum) [\(get / set\)](#)

設定値 :

AvsDirectionType (enum 値)	値	説明
ADT_L2R	0	左綴じ (Left to Right)
ADT_R2L	1	右綴じ (Right to Left)

解説：ページレイアウトが見開き表示時の綴じ方です。

PDF ファイルに指定がある場合はその設定に従って表示されます。指定がない場合のデフォルトは「左綴じ」です。表示上の設定は変更できますが、PDF ファイルには反映されません。

#### 5.5.18. CtrlStyle プロパティ

---

型 : int32 [\(.NET\)](#) [\(get / set\)](#)

設定値 :

AvsCtrlStyle	値	説明
AGCS_HIDEOBJECTS	0x00000001	注釈等のオブジェクトを表示しません
AGCS_PREVIEWMODE	0x00000002	プレビューモード（編集不可、テキスト選択不可、注釈選択不可、ポップアップ注釈非表示）に設定します
AGCS_HIDEERRDLG	0x00000004	エラーダイアログを表示しません
AGCS_HIDEPASSWORDDLG	0x00000008	パスワード入力ダイアログを表示しません
AGCS_READONLY	0x00000010	読み専用モード（編集不可）に設定します
AGCS_IGNOREOPENPAGELAYOUT	0x00000020	PDFに設定されたオープンページレイアウトを無視します。
AGCS_IGNOREOPENACTION	0x00000040	PDFに設定されたオープンアクションを無視します。
AGCS_HIDEANNOTATIONS	0x00000080	注釈を非表示にします。
AGCS_NOTURNATSINGLELAYOUT	0x00000100	単ページ表示のとき、キーまたはマウスホイールでのスクロールによるページ遷移を抑制します。
AGCS_DONTONKILLFOCUSBWND	0x00000200	テキスト入力がフォーカスを失ったときに再度フォーカスを当てるさせません。
AGCS_DONTOFFSAHIGHLIGHTS	0x00000400	ページ上をクリックしたとき、全検索結果のハイライトが消えないようにします。
AGCS_DISABLECURSOR	0x00000800	コントロールのマウスカーソルの制御を抑止します。  ※注釈ポップアップの入力欄、テキストボックス・引き出し線・タイプライターのテキスト入力、PDFPageViewCtrlで表示するWindows標準部品は抑止されません。
AGCS_SHOWGRID	0x00040000	グリッドを表示します。
AGCS_DISABLEDRAGCOPY	0x00100000	CTRL+Dragによる注釈コピーを禁止します。
AGCS_DISABLEALTTXTSEL	0x00200000	ALT+ドラッグによる範囲指定テキスト選択を禁止します。

解説：コントロールのスタイルを上記ビットの組み合わせで設定します。デフォルトは 0 です。0x0011 を指定（読み専用、オブジェクト非表示）すると V2.5 相当になります。

(V5 以降) AGCS\_ALIGNVERTICALCENTER は廃止されました。

### 5.5.19. HasUserPassword プロパティ

---

型： bool (.NET) (get のみ)

解説：PDF にユーザパスワードが設定されているか調べます。ユーザパスワードが設定されていれば True になります。

通常はオープン時の PDF の状態を返しますが、UserPassword プロパティに値をセットした場合、状態が変更されます。ドキュメントをオープンしていない場合は必ず False になります。

### 5.5.20. HasOwnerPassword プロパティ

---

型： bool (.NET) (get のみ)

解説：PDF にオーナパスワードが設定されているか調べます。オーナパスワードが設定されていれば True になります。

通常はオープン時の PDF の状態を返しますが、OwnerPassword プロパティに値をセットした場合、状態が変更されます。ドキュメントをオープンしていない場合は必ず False になります。

### 5.5.21. UserPassword プロパティ

---

型： string (.NET) (get / set)

解説：ユーザパスワードです。オープン時にユーザパスワードを指定して PDF を開くとその値が入ります。空文字をセットするとパスワードを設定していないことになり、HasUserPassword で False が返るようになります。値のセットは IsLockedByOwnerPassword() が True の状態では失敗します。

### 5.5.22. OwnerPassword プロパティ

---

型： string (.NET) (get / set)

解説：オーナーパスワードです。オープン時にオーナーパスワードを指定して PDF を開くとその値が入ります。空文字をセットするとパスワードを設定していないことになり、HasOwnerPassword で False が返るようになります。

値のセットは IsLockedByOwnerPassword() が True の状態では失敗します。

#### 5.5.23. SecurityPermissionFlag プロパティ

---

型： int32 (.NET) (get / set)

解説：PDF のセキュリティ許可フラグです。

次のビットの組み合わせとなります。

AvsSecurityPermissionFlag	値	説明
ASPF_ENABLE_PRINT	0x0004	印刷許可
ASPF_ENABLE_CHANGEDOC	0x0008	ドキュメント内容の変更許可
ASPF_ENABLE_COPYCONTENT	0x0010	内容のコピー許可
ASPF_ENABLE_MODIFYANNOT	0x0020	注釈の編集許可
ASPF_ENABLE_FILLFORM	0x0100	フォームフィールドの入力許可
ASPF_ENABLE_ACCESSIBILITY	0x0200	アクセシビリティの許可
ASPF_ENABLE_ASSEMBLEDDOC	0x0400	ページの挿入、削除、回転許可
ASPF_ENABLE_PRINTHIGHLEVEL	0x0804	高解像度の印刷許可

値のセットは IsLockedByOwnerPassword が True の状態では失敗します。

#### 5.5.24. ZoomRatio プロパティ

---

型： float32 (.NET) (get / set)

解説：現在のズーム倍率です。値をセットすると ViewMode が自動的に 0 になります。

#### 5.5.25. ZoomRatioLimit プロパティ

---

型： float32 (.NET) (get のみ)

解説：最大ズーム倍率を取得します。

#### 5.5.26. CurrentPageNumber プロパティ

---

型： int32 (.NET) (get のみ)

解説：現在のページ番号です。

#### 5.5.27. CanUndo プロパティ

---

型： bool (.NET) (get のみ)

解説：Undo ができるかどうか調べます。Undo 可能ならば True になります。

#### 5.5.28. CanRedo プロパティ

---

型： bool (.NET) (get のみ)

解説：Redo ができるかどうか調べます。Redo 可能ならば True になります。

#### 5.5.29. CanCut プロパティ

---

型： bool (.NET) (get のみ)

解説：Cut ができるかどうか調べます。Cut 可能ならば True になります。

#### 5.5.30. CanCopy プロパティ

---

型： bool (.NET) (get のみ)

解説：Copy ができるかどうか調べます。Copy 可能ならば True になります。

本文テキストを選択している状態でも PDF にコピー不可のセキュリティがかかっていた場合、false が返ります。

### 5.5.31. CanPaste プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： Paste ができるかどうか調べます。Paste 可能ならば True になります。

### 5.5.32. CanDelete プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： Delete ができるかどうか調べます。Delete 可能ならば True になります。

### 5.5.33. CanModifyPage プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： ページ編集（回転）ができるかどうか調べます。ページ編集可能ならば True になります。

### 5.5.34. CanModifyAnnotation プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： 注釈の編集（追加、削除、変更）ができるかどうか調べます。注釈編集可能ならば True になります。

### 5.5.35. CanModifyFontStyle プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： フォントスタイル変更ができるかどうか調べます。フォントスタイル変更可能ならば True になります。

### 5.5.36. CanUndoView プロパティ

---

型： bool (.NET) (getのみ)

解説： UndoView ができるかどうか調べます。 UndoView 可能ならば True になります。

### 5.5.37. CanRedoView プロパティ

---

型： bool (.NET) (getのみ)

解説： RedoView ができるかどうか調べます。 RedoView 可能ならば True になります。

### 5.5.38. CanPrint プロパティ

---

型： bool (.NET) (getのみ)

解説： PDF に印刷制御のセキュリティがかかっているか調べます。 印刷可能ならば True になります。 印刷不可の場合、印刷はおこなえません。

### 5.5.39. SelectedObjectCount プロパティ

---

型： int32 (.NET) (getのみ)

解説： 選択されているオブジェクトの数を取得します。

### 5.5.40. PdfVersion プロパティ

---

型： string (.NET) (getのみ)

解説： PDF のバージョンです。

#### 5.5.41. CreationDate プロパティ

---

型： string (.NET) (getのみ)

解説：PDF メタ情報の作成日時です。

#### 5.5.42. ModifyDate プロパティ

---

型： string (.NET) (getのみ)

解説：PDF メタ情報の更新日時です。

#### 5.5.43. Title プロパティ

---

型： string (.NET) (get / set)

解説：PDF メタ情報のタイトルです。

値のセットは IsLockedByOwnerPassword() が true の状態では失敗します。

#### 5.5.44. Author プロパティ

---

型： string (.NET) (get / set)

解説：PDF メタ情報の作成者名です。

値のセットは IsLockedByOwnerPassword() が true の状態では失敗します。

#### 5.5.45. Subject プロパティ

---

型： string (.NET) (get / set)

解説：PDF メタ情報のサブタイトルです。

値のセットは IsLockedByOwnerPassword() が true の状態では失敗します。

#### 5.5.46. Keywords プロパティ

---

型： string (.NET) (get / set)

解説：PDF メタ情報のキーワードです。

値のセットは IsLockedByOwnerPassword() が true の状態では失敗します。

(V5 以降) 複数行を指定する場合改行は"¥r¥n"を使用します。

#### 5.5.47. Creator プロパティ

---

型： string (.NET) (get のみ)

解説：PDF メタ情報の作成したアプリケーション名です。

#### 5.5.48. Producer プロパティ

---

型： string (.NET) (get のみ)

解説：PDF メタ情報の PDF 変換をおこなったツール名です。

#### 5.5.49. DocumentPath プロパティ

---

型： string (.NET) (get のみ)

解説：オープン中 PDF ファイルの Path を取得します。

V5.0 以降では SaveDocument でファイルパスを指定した後もオープンしたパスを返します。

#### 5.5.50. SDKVersion プロパティ

---

型： string (.NET) (get のみ)

解説：PDFViewerSDK のバージョン番号です。

#### 5.5.51. SearchStartPage プロパティ

---

型： int32 (.NET) (get / set)

解説：検索範囲の開始ページ番号です。指定された開始ページと最終ページの範囲で検索が行われます。Search メソッドや ShowSearchDialog メソッドで検索を開始する前に指定してください。指定されたページ番号が最終ページ番号より大きい場合は最終ページが検索の開始ページになります。

#### 5.5.52. LoadPageCount プロパティ

---

型： int32 (.NET) (get / set)

解説：メモリに読み込むページ数を設定します。設定のみで取得はできません。設定はオープンの前に行ってください。

設定したページ数より PDF のページ数が多いと、設定したページ数までメモリに持ちます。

初期値は 200 です。値の範囲は 2 以上、上限はありません。

ファイル名でオープンした場合はファイルがオープンしたままになります。オープンしたままにしたくない場合はストリームでオープンしてください。

※メモリを多く消費する PDF でメモリ不足が発生する場合に使用します。通常は指定する必要はありません。

#### 5.5.53. SelectedTextString プロパティ

---

型： string (.NET) (get のみ)

解説： 選択中のテキストを取得します。

テキストが選択されていない場合は、空文字列になります。

#### 5.5.54. ViewCoord プロパティ

---

型： bool (.NET) (get / set)

解説： 現在の座標系がビュー座標系であるかを取得および設定します。

False の場合、従来の論理座標系になります。

ビュー座標系および論理座標系について「4.4 座標系について」をご覧ください。

#### 5.5.55. IncludeArtifact プロパティ

---

型： bool (.NET) (get / set)

解説： API や GUI 操作の結果にテキスト装飾（タグ付き PDF において”Artifact”とタグ付けされたテキスト）を含めるかを取得・設定します。True の場合含めます。

指定は PDF オープン前に行ってください。オープン後のプロパティの設定は反映されません。

対象となるメソッドや操作は以下になります。

GetTextInfoInRect, GetSelectedTextInfo, HitTest, SelectedTextString

テキスト選択、マウスカーソル（テキスト検索は対象外です）

#### 5.5.56. SnapPointType プロパティ

---

型： int (.NET) (get / set)

設定値：

AvsSntapPointType	値	説明
APPT_END	0x00000001	終点
APPT_MIDDLE	0x00000002	中間点
APPT_CROSS	0x00000004	交点
APPT_ONLINE	0x00000008	線上

解説： ものさしでの測定時にスナップする点の種類を取得または設定します。

AvsSnapPointType の組み合わせを設定してください。

#### 5.5.57. HasRedactAnnotation プロパティ

---

型： bool (.NET) (get)

解説： オープン中のドキュメントが Redact 注釈を含むかを返します。

True の場合、Redact 注釈を含みます。False の場合、含みません。

#### 5.5.58. ScrollPosition プロパティ

---

型： LogicalPosition [\(.NET\)](#) [\(get\)](#)

解説：現在のスクロール位置をあらわす論理座標を返します。

#### 5.5.59. DeviceScrollPosition プロパティ

---

型： System.Drawing.Point [\(.NET\)](#) [\(get\)](#)

解説：現在のスクロール位置をあらわすクライアント座標を返します。

#### 5.5.60. OpenDocument メソッド

---

**int32 OpenDocument(string filePath, string password); [\(.NET\)](#)**

filePath : PDF ファイルのパスを指定します。

password : パスワード パスワード無しの場合は空文字列を指定してください。

戻り値：成功した場合 0 が返ります。失敗した場合はエラーコードが返ります。

解説： PDF ファイルをオープンします。 閲覧パスワードがかかっている PDF の場合、デフォルトではパスワード入力ダイアログを表示します。それ以外のエラーが発生した場合はエラーダイアログを表示します。

エラー時にダイアログを一切表示したくない場合は CtrlStyle プロパティで 0x0004 を 指定してください。また、0x0008 を指定するとパスワードがかかった PDF でもパスワード入力ダイアログが表示されません。この場合パスワード処理は自分でおこなう必要があります。エラーコードが 843906 の場合は、PDF にパスワードが設定されていて、与えられた password が一致しないエラーです。パスワード入力をを行い、再度 OpenDocument() を呼び出してください。

#### 5.5.61. SaveDocument メソッド

---

**int32 SaveDocument(string filePath); [\(.NET\)](#)**

filePath : PDF ファイルのパスを指定します。

戻り値：成功した場合 0 が返ります。失敗した場合はエラーコードが返ります。

解説： PDF ファイルを保存します。

エラー時にダイアログを一切表示したくない場合は CtrlStyle プロパティで 0x0004 を指定してください。

### 5.5.62. CloseDocument メソッド

---

**void CloseDocument();** (.NET)

解説：PDF ファイルをクローズします。

### 5.5.63. PrintDocument メソッド

---

```
bool PrintDocument(bool bShowDialog, string printerName, int32 fromPageNo, int32 toPageNo,  
int32 paperSize, int32 paperWidth, int32 paperHeight, AvsPrintOrientation orientation,  
AvsPrintPageSizeMode pageSizeMode, bool isCenterAlign, int32 copies,  
bool isCollate, string jobName, int32 exMode); (.NET)
```

bShowDialog : 印刷ダイアログを表示する場合 True を指定します。

printerName : プリンタ名

fromPageNo : 開始ページ番号。 0 を指定すると 1 ページになります。

toPageNo : 終了ページ番号。 0 を指定すると最終ページになります。

paperSize : 用紙サイズ。 0~118,256 を指定できます。0 を指定するとプリンタのデフォルト用紙サイズになります。※解説参照

paperWidth : 用紙幅(0.1mm 単位)。用紙サイズに 256 を指定した場合のみ有効です。

paperHeight : 用紙の高さ(0.1mm 単位)。用紙サイズに 256 を指定した場合のみ有効です。

orientation : 用紙の向きを次の値で指定します。

AvsPrintOrientation	値	説明
APO_AUTO	0	ページサイズに合わせて向きを変えます
APO_PORTRAIT	1	縦
APO_LANDSCAPE	2	横

pageSizeMode : ページの拡大・縮小を次の値で指定します。

AvsPrintPageSizeMode	値	説明
APS_NORMAL	0	なし
APS_FITPAPER	1	用紙に合わせて拡大、縮小する
APS_SHRINKTOPAPER	2	大きいページを縮小

isCenterAlign : 用紙の中央に印刷する場合は True を指定します。

copies : 印刷部数。

isCollate : 部単位印刷をおこなう場合は True を指定します。

jobName : プリンタのジョブ名。省略するとファイルパス（ストリームの場合は"PDF File"）になります。

exMode : 下記ビットの組み合わせで指定します。

AvsPrintExMode	値	説明
APM_DUPLEX	0x0001	両面印刷 (APM_DUPHORZ を指定する と短辺とじ、指定しないと長辺とじ)
APM_DUPHORZ	0x0002	短辺とじ。APM_DUPLEX の指定が必要 です。

戻り値：成功した場合 true が返ります。失敗した場合は false が返ります。

解説： 印刷をおこないます。ダイアログを表示する場合は bShowDialog 以外の引数はダイアログのデフォルト値として使用されます。PDF に印刷不可のセキュリティがかかっている場合は印刷できません。

指定したプリンタに対する印刷権限がない場合はデフォルトプリンタに印刷されます。

paperSize には次の値を設定できます。

- 0 プリンタのデフォルト用紙サイズ
- 1 Letter 8 1/2 x 11 in
- 2 Letter Small 8 1/2 x 11 in
- 3 Tabloid 11 x 17 in
- 4 Ledger 17 x 11 in
- 5 Legal 8 1/2 x 14 in
- 6 Statement 5 1/2 x 8 1/2 in
- 7 Executive 7 1/4 x 10 1/2 in
- 8 A3 297 x 420 mm
- 9 A4 210 x 297 mm
- 10 A4 Small 210 x 297 mm
- 11 A5 148 x 210 mm
- 12 B4 (JIS) 250 x 354
- 13 B5 (JIS) 182 x 257 mm
- 14 Folio 8 1/2 x 13 in
- 15 Quarto 215 x 275 mm
- 16 10x14 in
- 17 11x17 in
- 18 Note 8 1/2 x 11 in
- 19 Envelope #9 3 7/8 x 8 7/8
- 20 Envelope #10 4 1/8 x 9 1/2
- 21 Envelope #11 4 1/2 x 10 3/8
- 22 Envelope #12 4 1/2 x 10 3/8
- 23 Envelope #14 5 x 11 1/2

- 24 C size sheet
- 25 D size sheet
- 26 E size sheet
- 27 Envelope DL 110 x 220mm
- 28 Envelope C5 162 x 229 mm
- 29 Envelope C3 324 x 458 mm
- 30 Envelope C4 229 x 324 mm
- 31 Envelope C6 114 x 162 mm
- 32 Envelope C65 114 x 229 mm
- 33 Envelope B4 250 x 353 mm
- 34 Envelope B5 176 x 250 mm
- 35 Envelope B6 176 x 125 mm
- 36 Envelope 110 x 230 mm
- 37 Envelope Monarch 3.875 x 7.5 in
- 38 6 3/4 Envelope 3 5/8 x 6 1/2 in
- 39 US Std Fanfold 14 7/8 x 11 in
- 40 German Std Fanfold 8 1/2 x 12 in
- 41 German Legal Fanfold 8 1/2 x 13 in
- 42 B4 (ISO) 250 x 353 mm
- 43 Japanese Postcard 100 x 148 mm
- 44 9 x 11 in
- 45 10 x 11 in
- 46 15 x 11 in
- 47 Envelope Invite 220 x 220 mm
- 48 RESERVED--DO NOT USE
- 49 RESERVED--DO NOT USE
- 50 Letter Extra 9 ¥275 x 12 in
- 51 Legal Extra 9 ¥275 x 15 in
- 52 Tabloid Extra 11.69 x 18 in
- 53 A4 Extra 9.27 x 12.69 in
- 54 Letter Transverse 8 ¥275 x 11 in
- 55 A4 Transverse 210 x 297 mm
- 56 Letter Extra Transverse 9¥275 x 12 in
- 57 SuperA/SuperA/A4 227 x 356 mm
- 58 SuperB/SuperB/A3 305 x 487 mm
- 59 Letter Plus 8.5 x 12.69 in
- 60 A4 Plus 210 x 330 mm
- 61 A5 Transverse 148 x 210 mm
- 62 B5 (JIS) Transverse 182 x 257 mm
- 63 A3 Extra 322 x 445 mm
- 64 A5 Extra 174 x 235 mm
- 65 B5 (ISO) Extra 201 x 276 mm
- 66 A2 420 x 594 mm
- 67 A3 Transverse 297 x 420 mm

- 68 A3 Extra Transverse 322 x 445 mm
- 69 Japanese Double Postcard 200 x 148 mm
- 70 A6 105 x 148 mm
- 71 Japanese Envelope Kaku #2
- 72 Japanese Envelope Kaku #3
- 73 Japanese Envelope Chou #3
- 74 Japanese Envelope Chou #4
- 75 Letter Rotated 11 x 8 1/2 11 in
- 76 A3 Rotated 420 x 297 mm
- 77 A4 Rotated 297 x 210 mm
- 78 A5 Rotated 210 x 148 mm
- 79 B4 (JIS) Rotated 364 x 257 mm
- 80 B5 (JIS) Rotated 257 x 182 mm
- 81 Japanese Postcard Rotated 148 x 100 mm
- 82 Double Japanese Postcard Rotated 148 x 200 mm
- 83 A6 Rotated 148 x 105 mm
- 84 Japanese Envelope Kaku #2 Rotated
- 85 Japanese Envelope Kaku #3 Rotated
- 86 Japanese Envelope Chou #3 Rotated
- 87 Japanese Envelope Chou #4 Rotated
- 88 B6 (JIS) 128 x 182 mm
- 89 B6 (JIS) Rotated 182 x 128 mm
- 90 12 x 11 in
- 91 Japanese Envelope You #4
- 92 Japanese Envelope You #4 Rotated
- 93 PRC 16K 146 x 215 mm
- 94 PRC 32K 97 x 151 mm
- 95 PRC 32K(Big) 97 x 151 mm
- 96 PRC Envelope #1 102 x 165 mm
- 97 PRC Envelope #2 102 x 176 mm
- 98 PRC Envelope #3 125 x 176 mm
- 99 PRC Envelope #4 110 x 208 mm
- 100 PRC Envelope #5 110 x 220 mm
- 101 PRC Envelope #6 120 x 230 mm
- 102 PRC Envelope #7 160 x 230 mm
- 103 PRC Envelope #8 120 x 309 mm
- 104 PRC Envelope #9 229 x 324 mm
- 105 PRC Envelope #10 324 x 458 mm
- 106 PRC 16K Rotated
- 107 PRC 32K Rotated
- 108 PRC 32K(Big) Rotated
- 109 PRC Envelope #1 Rotated 165 x 102 mm
- 110 PRC Envelope #2 Rotated 176 x 102 mm
- 111 PRC Envelope #3 Rotated 176 x 125 mm

112 PRC Envelope #4 Rotated 208 x 110 mm  
113 PRC Envelope #5 Rotated 220 x 110 mm  
114 PRC Envelope #6 Rotated 230 x 120 mm  
115 PRC Envelope #7 Rotated 230 x 160 mm  
116 PRC Envelope #8 Rotated 309 x 120 mm  
117 PRC Envelope #9 Rotated 324 x 229 mm  
118 PRC Envelope #10 Rotated 458 x 324 mm  
256 カスタムサイズ paperWidth/ paperHeight に任意の値を指定してください。

#### 5.5.64. GetPDFDocumentHandle メソッド

---

**IntPtr GetPDFDocumentHandle();** [\(.NET\)](#)

戻り値： PDFDocument のハンドル

解説：コントロールに関連付けられている PDFDocument のハンドルを取得します。ハンドルは OpenDocument でファイルを開いた後に取得してください。

#### 5.5.65. AttachPDFDocument メソッド

---

**void AttachPDFDocument(IntPtr hPDFDocument);** [\(.NET\)](#)

hPDFDocument : PDFDocument のハンドル

解説：指定した PDFDocument のハンドルをコントロールに関連付けます。一つの PDFDocument を、複数のコントロールと使い回したいときに使用します。同じ PDFDocument を持つコントロールはアンドゥー、回転、ファイルクローズ時などを同期します。

#### 5.5.66. UnlockOwnerPasswordSecurity メソッド

---

**bool UnlockOwnerPasswordSecurity (string password);** [\(.NET\)](#)

password : オーナーパスワードを指定します。

戻り値：セキュリティ解除に成功した場合、true が返ります。

解説：オーナーパスワードを指定してセキュリティを解除します。

PDF にオーナパスワードがかかっている場合、オープン時にオーナパスワードを入力して PDF を開くか、この関数でセキュリティを解除しない限り、ページの回転、オブジェクトの追加、削除、変更といった編集機能は一切動作しません。

#### 5.5.67. GetPageWidth メソッド

---

**int32 GetPageWidth(int32 pageNumber); (.NET)**

pageNumber : ページ番号を指定します。

戻り値 : 指定したページの幅が返ります。 (単位 : twip)

解説 : 指定したページの幅を取得します。単位は Twip です。PDF がオープンされていない場合は 0 になります。

#### 5.5.68. GetPageHeight メソッド

---

**int32 GetPageHeight(int32 pageNumber); (.NET)**

pageNumber : ページ番号を指定します。

戻り値 : 指定したページの高さが返ります。 (単位 : twip)

解説 : 指定したページの高さを取得します。単位は Twip です。PDF がオープンされていない場合は 0 になります。

#### 5.5.69. GetNextZoomRatio メソッド

---

**float32 GetNextZoomRatio (bool bZoomIn); (.NET)**

bZoomIn : 拡大時の次ズーム倍率を取得する場合は True、縮小時は False を指定します。

戻り値 : ズーム倍率

解説 : ズーム倍率の次の刻み値を取得します。刻み値はコントロール内部で決められた値で下から順に 10 → 25 → 50 → 60 → 75 → 90 → 100 → 125 → 150 → 200 → 300 → 400 → 600 → 800 → 1200 → 1600 → 2400 → 3200 → 6400 になります。

マーキーズームやマウスホイールによる拡大／縮小はこの刻み値に従っておこなわれます。

## 5.5.70. GotoPage メソッド

---

**bool GotoPage (int32 pageNo);** (.NET)

戻り値：表示するページ

戻り値：True 成功（ページが切り替わった場合）

解説：指定したページを表示します。

## 5.5.71. GotoFirstPage メソッド

---

**bool GotoFirstPage ();** (.NET)

戻り値：true 成功（ページが切り替わった場合）

解説：先頭ページを表示します。

## 5.5.72. GotoLastPage メソッド

---

**bool GotoLastPage ();** (.NET))

戻り値：True 成功（ページが切り替わった場合）

解説：最終ページを表示します。

## 5.5.73. GotoPreviousPage メソッド

---

**bool GotoPreviousPage ();** (.NET))

戻り値：True 成功（ページが切り替わった場合）

解説：前ページを表示します。

## 5.5.74. GotoNextPage メソッド

---

**bool GotoNextPage ();** (.NET)

戻り値：True 成功（ページが切り替わった場合）

解説：次ページを表示します。

#### 5.5.75. GetPageRotation メソッド

---

**int32 GetPageRotation (int32 pageNo); ([.NET](#))**

pageNo : ページ番号

戻り値：回転角度

解説：指定したページの回転角度を取得します。

#### 5.5.76. RotatePages メソッド

---

**void RotatePages (int32 pageNo, int32 rotation); ([.NET](#))**

pageNo : ページ番号 -1 を指定すると全ページ

rotation : 回転角度。90/180/270 のいずれかを指定

解説：指定したページを回転します。回転は 90 度単位です。

#### 5.5.77. ResetView メソッド

---

**void ResetView(); ([.NET](#))**

解説：ビューを一度クリアして作り直します。

#### 5.5.78. Undo メソッド

---

**void Undo(); ([.NET](#))**

解説：最後の編集操作を取り消します。最高 50 回まで可能です。CanUndo()でアンドゥー可能かどうか調べることができます。

### 5.5.79. Redo メソッド

---

**void Redo();** [\(.NET\)](#)

解説：Undo を取り消します。最高 50 回まで可能です。CanRedo() でリドゥー可能かどうか調べることができます。

### 5.5.80. Cut メソッド

---

**void Cut();** [\(.NET\)](#)

解説：選択されたオブジェクトやテキストボックスの文字列を切り取ってクリップボードに保存します。CanCut() で切り取り可能かどうか調べることができます。

### 5.5.81. Copy メソッド

---

**void Copy();** [\(.NET\)](#)

解説：選択された本文テキスト、オブジェクト、テキストボックスの文字列をコピーしてクリップボードに保存します。CanCopy() でコピー可能かどうか調べることができます

### 5.5.82. Paste メソッド

---

**void Paste();** [\(.NET\)](#)

解説：クリップボードの内容（Cut や Copy したオブジェクト、文字列）を貼り付けます。CanPaste() で貼り付け可能かどうか調べることができます。

### 5.5.83. Delete メソッド

---

**void Delete();** [\(.NET\)](#)

解説：選択されたオブジェクトやテキストボックスの文字列を削除します。CanDelete()で削除可能かどうか調べることができます。

#### 5.5.84. SelectAll メソッド

---

**void SelectAll(bool bSelect);** [\(.NET\)](#)

bSelect : True を指定すると選択、False を指定すると選択解除

解説：ページテキストをすべて選択、または選択解除します。

#### 5.5.85. ShowDocumentPropertyDialog メソッド

---

**void ShowDocumentPropertyDialog();** [\(.NET\)](#)

解説：PDF ドキュメントのプロパティ設定ダイアログを表示します。

#### 5.5.86. ShowFontStyleDialog メソッド

---

**void ShowFontStyleDialog();** [\(.NET\)](#)

解説：フリーテキスト注釈のフォントスタイル設定ダイアログを表示します。フリーテキスト注釈やその中のテキストを選択している状態で使用することができます。使用可能な状態かどうかは CanModifyFontStyle で調べることができます。

#### 5.5.87. ShowObjectPropertyDialog メソッド

---

**void ShowObjectPropertyDialog();** [\(.NET\)](#)

解説：選択されているオブジェクトのプロパティ設定ダイアログを表示します。オブジェクトが一つ以上選択されている必要があります。

#### 5.5.88. UndoView メソッド

---

**void UndoView();** ([.NET](#))

解説：表示をアンドゥーして一つ前の表示に戻します。最高 50 回まで可能です。CanUndoView プロパティでアンドゥー可能かどうか調べることができます。

#### 5.5.89. RedoView メソッド

---

**void RedoView();** ([.NET](#))

解説：UndoView を取り消します。最高 50 回可能です。CanRedoView プロパティでリドゥー可能かどうか調べることができます。

#### 5.5.90. ShowSearchDialog メソッド

---

**void ShowSearchDialog(string searchStr, int32 bMatchCase, int32 bSearchAll);** ([.NET](#))

searchStr : 検索文字列。NULL を指定すると選択中の文字列がある場合はその文字列、ない場合は前回と同じ文字列になります。

bMatchCase : 大文字と小文字を区別する場合は 1 を、区別しない場合は 0 を、前回と同じ指定にする場合は 2 を指定します。

bSearchAll : 全検索する場合は 1 を、しない場合は 0 を、前回と同じ指定にする場合は 2 を指定します。

解説：検索ダイアログを表示して検索をおこないます。3つの引数は検索ダイアログ表示時の初期値となります。

#### 5.5.91. Search メソッド

---

**bool Search(string searchStr, bool bMatchCase, bool bSearchAll, bool bReverse, bool bMultiStr);** ([.NET](#))

searchStr : 検索文字列。NULL を指定すると選択中の文字列がある場合はその文字列、ない場合は前回と同じ文字列になります。

bMatchCase : 大文字と小文字を区別する場合は True を指定します。

bSearchAll : 全検索する場合は True を、しない場合は False を指定します。

bReverse : リバース検索（逆方向に検索）をおこなう場合は True を指定します。

bMultiStr : 複数単語検索をおこなう場合は True を指定します。文字列の指定は searchStr にスペース区切りで複数の文字列をセットしてください。複数単語検索は全検索時のみ有効です。

戻り値 : True 成功。

解説：検索をおこないます。検索後は SearchNext、SearchPrevious が有効になります。検索に失敗したときは「検索文字列が見つかりませんでした。」というメッセージが表示されます。これを非表示にしたい場合は、CtrlStyle プロパティで **AGCS\_HIDEERRDLG** を指定してください。

(V5 以降) SearchNext や SearchPrevious メソッドを繰り返し呼び出し一巡した場合、再び、最初から検索を繰り返します。

#### 5.5.92. SearchNext メソッド

---

**void SearchNext();** [\(.NET\)](#)

解説：次検索をおこないます。

#### 5.5.93. SearchPrevious メソッド

---

**void SearchPrevious();** [\(.NET\)](#)

解説：前検索をおこないます。

#### 5.5.94. ClearSelect メソッド

---

**void ClearSelect();** [\(.NET\)](#)

解説：選択中のテキストをクリア（非選択状態）にします。検索によって選択状態になったテキストもクリアされます。

#### 5.5.95. MakeBmpPage メソッド

---

**bool MakeBmpPage(int32 pageNo, int32 dpi, string scale, int32 height, AvsImageColor nColor,**  
**System.Drawing.Rectangle rcClip, string filePath);** [\(.NET\)](#)

戻り値：成功の場合は True、失敗したら False を返します。

pageNo : ページ番号

dpi : スケーリング時の DPI 値(1 以上、1440 以下)

scale : 出力時のスケール文字列。100%以下、"1"=="100%","100px"は幅の指定

height : 出力イメージの最大高さ(Pixel)

nColor : 色数を次の値で指定します。

AvsImageColor	値	説明
IMC_COLOR	0	フルカラー
IMC_256COLOR	1	256 色カラー
IMC_GRAYSCALE	2	グレースケール
IMC_MONOCHROME	3	モノクローム

clipLeft/clipTop/clipRight/clipBottom/rcClip : イメージを作成するページ内の領域。空の矩形はページ全体  
(twip)

filePath : 出力先ファイルパス

解説：指定ページを BMP 画像に変換します。

※ 注釈等のオブジェクトは外観ストリーム情報をもつものだけが表示されるためページビューの表示とは異なる場合があります。また編集内容は反映されません。

## 5.5.96. MakeJpegPage メソッド

---

```
bool MakeJpegPage(int32 pageNo, int32 dpi, string scale, int32 height, AvsImageColor nColor,  
int32 quality, System.Drawing.Rectangle rcClip, string filePath); (.NET)
```

戻り値：成功の場合は True、失敗したら False を返します。

pageNo : ページ番号

dpi : スケーリング時の DPI 値(1 以上、1440 以下)

scale : 出力時のスケール文字列。100%以下、"1"=="100%","100px"は幅の指定

height : 出力イメージの最大高さ(Pixel)

nColor : 色数を指定します。[MakeBmpPage の同変数を参照してください。](#)

quality : 変換品質 (1-100 %) 1-100 以外の場合は 80 として処理します。

rcClip : イメージを作成するページ内の領域。空の矩形はページ全体(twip)

filePath : 出力先ファイルパス

解説：指定ページを JPEG 画像に変換します。

※ 注釈等のオブジェクトは外観ストリーム情報をもつものだけが表示されるためページビューの表示とは異なる場合があります。また編集内容は反映されません。

#### 5.5.97. MakePngPage メソッド

---

```
bool MakePngPage(int32 pageNo, int32 dpi, string scale, int32 height, AvsImageColor nColor,  
System.Drawing.Rectangle rcClip, string filePath); (.NET)
```

戻り値：成功の場合は True、失敗したら False を返します。

pageNo : ページ番号

dpi : スケーリング時の DPI 値(1 以上、1440 以下)

scale : 出力時のスケール文字列。100%以下、"1"=="100%","100px"は幅の指定

height : 出力イメージの最大高さ(Pixel)

nColor : 色数を指定します。[MakeBmpPage の同変数を参照してください。](#)

rcClip : イメージを作成するページ内の領域。空の矩形はページ全体(twip)

filePath : 出力先ファイルパス

解説：指定ページを PNG 画像に変換します。注釈等オブジェクトの編集内容は反映されません。

#### 5.5.98. MakeTiffPage メソッド

---

```
bool MakeTiffPage(int32 startPageNo, int32 endPageNo, int32 dpi, string scale, int32 height,  
AvsImageColor nColor, int32 quality, System.Drawing.Rectangle rcClip,  
AvsTiffCompression compression, string filePath); (.NET)
```

戻り値：成功の場合は True、失敗したら False を返します。

startPageNo : 開始ページ番号

endPageNo : 終了ページ番号

dpi : スケーリング時の DPI 値(1 以上、1440 以下)

scale : 出力時のスケール文字列。100%以下、"1"=="100%","100px"は幅の指定

height : 出力イメージの最大高さ(Pixel)

nColor : 色数を指定します。[MakeBmpPage の同変数を参照してください。](#)

quality : DCT 変換品質 (1-100 %) compression で 2 の DCT が指定されたときのみ有効です。

1-100 以外の場合は 80 として処理します。

`rcClip` : イメージを作成するページ内の領域。空の矩形はページ全体(twip)

`compression` : 圧縮形式を次の値で指定します。デフォルトは 1 の LZW です。

AvsTiffCompression	値	説明
TCP_None	0	圧縮なし
TCP_LZW	1	LZW
TCP_DCT	2	DCT (JPEG)
TCP_Flate	3	Flate
TCP_RunLength	4	RunLength
TCP_CCITT3	5	CCITT3
TCP_CCITT4	6	CCITT4

`filePath` : 出力先ファイルパス

解説 : 指定ページを TIFF 画像に変換します。

※ 注釈等のオブジェクトは外観ストリーム情報をもつものだけが表示されるためページビューの表示とは異なる場合があります。また編集内容は反映されません。複数ページを指定した場合はページの編集内容（ページ移動等）も反映されません。

## 5.5.99. MakeSvgPage メソッド

**bool MakeSvgPage(int32 pageNo, System.Drawing.Rectangle rcClip, string filePath); (.NET)**

戻り値 : 成功の場合は True、失敗したら False を返します。

`pageNo` : ページ番号

`rcClip` : SVG を作成するページ内の領域。空の矩形はページ全体(twip)

`filePath` : 出力先ファイルパス

解説 : 指定ページを SVG ファイルに変換します。

※ 注釈等のオブジェクトは外観ストリーム情報をもつものだけが表示されるためページビューの表示とは異なる場合があります。また編集内容は反映されません。

## 5.5.100. GetDefaultObjectProperty/ GetDefaultObjectProperty2 メソッド

**AvsDotNetGuiCtl.ObjectProperty GetDefaultObjectProperty(AvsSelectMode type); (.NET)**

解説：指定したオブジェクトタイプのデフォルトプロパティを取得します。戻り値、引数の説明は [SetDefaultObjectProperty](#) を参照してください。

### 5.5.101. SetDefaultObjectProperty/ SetDefaultObjectProperty2 メソッド

---

**void SetDefaultObjectProperty(AvsSelectMode type, AvsDotNetGuiCtl.ObjectProperty op); (.NET)**

type : 設定するオブジェクトタイプ。※下記表参照

color : 色、境界線色を示す COLORREF 値 (-1 を指定すると透過)

fillColor : 背景色を示す COLORREF 値 (-1 を指定すると透過)

borderStyle : 境界線スタイル

AvsBorderStyle	値	説明
ABDS_SOLID	0	実線
ABDS_DOT	1	点線
ABDS_DASH1	2	破線 1
ABDS_DASH2	3	破線 2
ABDS_DASHDOT1	4	一点鎖線 1
ABDS_DASHDOT2	5	一点鎖線 2
ABDS_DASHDOT3	6	一点鎖線 3
ABDS_UNDERLINE	7	下線（リンク注釈のみ指定可）
ABDS_CLOUD1	8	雲型 1
ABDC_CLOUD2	9	雲型 2

borderWidth : 境界線の幅（単位：twip）

opacity : 不透明度(0~1.0)

lineCap1/lineCap2 : 始点/終点のラインキャップ

AvsLineCapType	値	説明
ALCT_NONE	0	なし
ALCT_OPENARROW	1	開いた矢印
ALCT_CLOSEDARROW	2	閉じた矢印
ALCT_ROPENARROW	3	開いた矢印（内向き）
ALCT_RCLOSEDARROW	4	閉じた矢印（内向き）
ALCT_BUTTON	5	突合わせ
ALCT_DIAMOND	6	ひし形

ALCT_CIRCLE	7	円形
ALCT_SQUARE	8	四角形
ALCT_SLASH	9	スラッシュ

fontName : フォント名

fontStyle : フォントスタイル 下記ビットの組み合わせを指定します。

AvsFontStyle	値	説明
AFTS_BOLD	0x0001	強調表示
AFTS_ITALIC	0x0002	斜体
AFTS_UNDERLINE	0x0004	下線
AFTS_STRIKEOUT	0x0008	取消線

fontSize : フォントサイズ (単位: ポイント)

fontColor : フォント色を示す COLORREF 値

text : 初期挿入するテキスト

exStyle : ノート注釈、ファイル添付ではアイコンの種類。リンクではハイライト表示スタイルを示します。

値	ノート注釈	ファイル添付	リンク
0	注釈	グラフ	なし
1	ヘルプ	クリップ	反転
2	テキスト挿入	添付	アウトライン
3	キー	タグ	プッシュ
4	新規段落		
5	ノート		
6	段落		

解説：指定したオブジェクトタイプのデフォルトプロパティを設定します。設定すると次回そのオブジェクトを新規作成したときにプロパティの初期値として使われます。

オブジェクトタイプによって設定可能なプロパティは異なります。（下記表参照）

AvsSelectMode	color	fillColor	border	border	border	opacity	line	line	font	font	font	font	text	exStyle	mask	Color
			Style	Width		Cap1	Cap2	Name	Style	Size	Color					
ASM_INS_STICKYNOTE	<input type="radio"/>				<input type="radio"/>								<input type="radio"/>			
ASM_INS_LINK			<input type="radio"/>	<input type="radio"/>									<input type="radio"/>			
ASM_INS_FILEATTACHMENT	<input type="radio"/>				<input type="radio"/>								<input type="radio"/>			
ASM_INS_TEXTBOX	<input type="radio"/>			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>									
ASM_INS_CALLOUT	<input type="radio"/>		<input type="radio"/>													
ASM_INS_TYPEWRITER								<input type="radio"/>								

ASM_INS_LINE	○	○	○	○	○	○	○							
ASM_INS_ARROW	○	○	○	○	○	○	○							
ASM_INS_RECTANGLE	○	○	○	○	○									
ASM_INS_OVAL	○	○	○	○	○									
ASM_INS_POLYGON	○	○	○	○	○									
ASM_INS_POLYLINE	○	○	○	○	○	○	○							
ASM_INS_HIGHLIGHT	○													
ASM_INS_UNDERLINE	○													
ASM_INS_CROSSOUT	○													
ASM_INS_SQUIGGLY	○													
ASM_INS_STAMP_APPROVED	○			○										
ASM_INS_STAMP_DONE	○			○										
ASM_INS_STAMP_TOPSECRET	○			○										
ASM_INS_STAMP_DRAFT	○			○										
ASM_INS_STAMP_FINAL	○			○										
ASM_INS_STAMP_EVALUATION	○			○										
ASM_INS_STAMP_PUBLIC	○			○										
ASM_INS_STAMP_REFERENCE	○			○										
ASM_INS_STAMP_DISMISSAL	○			○										
ASM_INS_STAMP_PRIVATE	○			○										
ASM_INS_STAMP_PRELIMINARY	○			○										
ASM_INS_STAMP_VOID	○			○										○
ASM_INS_STAMP_CUSTOM	○			○										
ASM_INS_LINE_DIMEMSION	○	○	○	○	○	○	○							
ASM_INS_POLYGON_DIMENSI ON	○	○	○	○	○									
ASM_INS_POLYLINE_DIMENSI ON	○	○	○	○	○	○	○							
ASM_INS_REDACT	○	○			○									
ASM_INS_CLOUD	○	○	○	○	○									

- ASM\_INS\_LINE\_DIMEMSION の場合は上記に加えて Leader/LeaderOffset/LeaderExt の指定が可能です。

AvsDotNetGuiCtl.ObjectProperty(.NET) はオブジェクトのプロパティをあらわすクラスです。

AvsDotNetGuiCtl.ObjectProperty メンバ		
型	メンバ名	説明
int32	Color	SetDefaultObjectProperty 引数を参照してください。
int32	FillColor	
AvsBorderStyle	BorderStyle	
float32	BorderWidth	
float32	Opacity	

AvsLineCapType	LineCap1	
AvsLineCapType	LineCap2	
String	FontName	
int32	FontStyle	
int32	FontSize	
int32	FontColor	
String	Text	
int32	ExStyle	
float32	Leader	
float32	LeaderOffset	
float32	LeaderExt	
int32	MaskColor	

### 5.5.102. GetCursorPosition メソッド

---

**AvsDotNetGuiCtl.LogicalPoint GetCursorPosition();** (.NET)

戻り値：論理座標（ページ番号、x, y）を返します。 (.NET)

解説：現在カーソル位置を論理座標（単位：twip）で取得します。

AvsDotNetGuiCtl.LogicalPoint(.NET) は論理座標をあらわすクラスです。

AvsDotNetGuiCtl.LogicalPoint クラス メンバ	
LogicalPoint(int32 pageNo, int32 x, int32 y)	コンストラクタ
int32 PageNumber	ページ番号
int32 X	X 座標
int32 Y	Y 座標

### 5.5.103. ScrollToPosition メソッド

---

**void ScrollToPosition(AvsDotNetGuiCtl.LogicalPoint /Pos, bool isAddUndoView);** (.NET)

/Pos : 論理座標 (.NET)

isAddUndoView : ビューのアンドゥを追加するかどうかの指定

解説：指定した論理位置がビューの左上にくるようにビューをスクロールします。AvsDotNetGuiCtl.LogicalPoint クラスについての説明は [GetCursorPosition](#) を参照してください。

#### 5.5.104. ScrollToDevicePosition メソッド

---

**void ScrollToDevicePosition(System.Drawing.Point point); (.NET)**

point : クライアント座標 [\(.NET\)](#)

解説：指定したクライアント位置がビューの左上にくるようにビューをスクロールします。

#### 5.5.105. ClientToLogical メソッド

---

**AvsDotNetGuiCtl.LogicalPoint ClientToLogical(System.Drawing.Point point); (.NET)**

戻り値 [\(.NET\)](#) : 論理座標（ページ番号、x, y）を返します。

point [\(.NET\)](#) : クライアント座標

解説：指定したクライアント座標を論理座標に変換します。

連続ページ表示や見開き表示では複数ページが同時に表示されるため、

論理座標と合わせて論理ページ番号が返ります。

※ページ領域外では、論理座標は負値や論理ページのページサイズを超える値となります。

#### 5.5.106. SetSelectionModeEx メソッド

---

**void.SetSelectionMode(AvsSelectionMode mode, int32 customOperation); (.NET)**

mode : カスタムモード（ASM\_CUSTOM1～ASM\_CUSTOM5）を指定します。詳細は [AvsSelectionMode](#) を確認してください。

customOperation : このカスタムモードに許可する操作を下記ビットの組み合わせで指定します。

AvsEnableOperation	値	説明
AEO_RCOBJSEL	0x0001	オブジェクトの矩形選択ができます。
AEO_TEXTSEL	0x0002	テキストの選択ができます。
AEO_LINKOP	0x0008	リンク注釈をクリックしてアクションを発動できます。
AEO_LINKEDT	0x0010	リンク注釈を選択、編集することができます。

AEO_ANNOTEDT	0x0020	リンク注釈を除く注釈を選択、編集することができます。
AEO_FORMOP	0x0040	フォームをクリックしてアクションを発動できます。 (現状は未対応)
AEO_FORMEDT	0x0080	フォームを選択、編集することができます。
AEO_DRAGSCRL	0x0100	ドラッグによるスクロールができます。
AEO_MQZOOM	0x0200	マーキーズーム機能を使うことができます。
AEO_PANSCRL	0x0400	タッチ操作によるスクロール、ピンチイン・ピンチアウトができます。

解説：独自のカスタム選択モードを作成、指定します。

#### 5.5.107. CreateObject メソッド

---

**String CreateObject(AvsDotNetGuiCtl.AvsSelectMode mode, int32 pageNo, long mask,  
AvsDotNetGuiCtl.AvsObject object); (.NET)**

戻り値：作成したオブジェクトの ID を返します。失敗した場合は空文字列を返します。

mode : 注釈作成モードを指定します。詳細は [AvsSelectMode](#) を確認してください。

pageNo : 注釈を作成するページ番号を指定します。

mask : どのパラメータを指定したいかを下記マスクビットの組み合わせで指定します。

AvsObjectMask	値	説明
AOPM_COLOR	0x00000001	色
AOPM_FILLCOLOR	0x00000002	背景色
AOPM_OPACITY	0x00000004	不透明度
AOPM_BORDERSTY LE	0x00000008	境界線スタイル
AOPM_BORDERWID TH	0x00000010	境界線幅
AOPM_LINECAP	0x00000020	始点/終点のラインキャップ
AOPM_FONTNAME	0x00000080	フォント名
AOPM_FONTSTYLE	0x00000100	フォントスタイル
AOPM_FONTSIZE	0x00000200	フォントサイズ
AOPM_FONTCOLOR	0x00000400	フォント色

AOPM_TEXT	0x00000800	テキスト
AOPM_EXSTYLE	0x00001000	<a href="#">特殊スタイル</a>
AOPM_COORDINATE	0x00010000	座標
AOPM_STATE	0x00020000	状態
AOPM_FILE	0x00040000	添付ファイル
AOPM SUBJECT	0x00080000	タイトル
AOPM_AUTHOR	0x00100000	作成者
AOPM_PARENTIDD	0x00200000	ポップアップ注釈の親注釈 ID
AOPM_LASTUPDATED	0x00400000	更新日時
AOPM_CREATIONDATE	0x00800000	作成日時
AOPM_LEADER	0x10000000	Leader/LeaderExt/LeaderOffset
AOPM_MASKCOLOR	0x20000000	マスクカラー
AOPM_ALL	0xFFFFFFFF	すべてのパラメータ

object : オブジェクトのパラメータを指定します。

#### AvsObject

AvsDotNetGuiCtl.AvsObject メンバ		
型	メンバ名	説明
<a href="#">AvsObjectType</a>	Type	オブジェクトタイプです。読み取り専用。
String	Id	オブジェクト ID です。読み取り専用。
Int32	State	オブジェクト状態のビットの組み合わせです。 <a href="#">AvsObjectState</a> 参照。
ArrayList	Coordinates	オブジェクトの論理座標です。 System.Drawing.Point の ArrayList 配列です。
<a href="#">AvsObjectProperty</a>	Property	オブジェクトのプロパティです。
String	FileName	添付ファイルのファイル名です。
Stream	FileBin	<ul style="list-style-type: none"> <li>・ 添付ファイルのバイナリです。</li> <li>・ カスタムスタンプの場合は画像ファイルのデータです。対応形式は bmp/gif/png/jpeg</li> </ul>

		です。（各形式の詳細仕様について対応していない場合がある可能性があります）
String	Subject	タイトルです。リンク注釈は取得、設定できません。
String	Author	作成者名です。リンク注釈は取得、設定できません。
String	ParentId	ポップアップ注釈の親注釈 ID です。読み取り専用。 ポップアップ注釈でのみ有効です。
String	LastUpdated	更新日時です。読み取り専用。
String	CreationDate	作成日時です。読み取り専用。
Int32	PageNumber	注釈が存在するページ番号。読み取り専用。(.Netのみ)

#### AvsObjectType

AvsObjectType	値	説明
OT_ANT_UNKNOWN	-1	不明
OT_ANT_TEXT	0	テキスト注釈（ノート注釈）
OT_ANT_LINK	1	リンク注釈
OT_ANT_FREETEXT	2	フリーテキスト注釈（テキストボックス、コールアウト、タイプライター）
OT_ANT_LINE	3	線注釈 / ものさし距離ツール
OT_ANT_SQUARE	4	四角形注釈
OT_ANT_CIRCLE	5	橢円注釈 / ものさし面積ツール
OT_ANT_POLYGON	6	多角形注釈 / ものさし周辺ツール
OT_ANT_POLYLINE	7	折れ線注釈
OT_ANT_HIGHLIGHT	8	ハイライト注釈
OT_ANT_UNDERLINE	9	下線注釈
OT_ANT_SQUIGGLY	10	波下線注釈
OT_ANT_STRIKEOUT	11	取消線注釈
OT_ANT_STAMP	12	スタンプ注釈
OT_ANT_CARET	13	キャレット注釈

OT_ANT_INK	14	インク注釈
OT_ANT_POPUP	15	ポップアップ注釈
OT_ANT_FILEATTACHMENT	16	ファイル添付注釈
OT_ANT_SOUND	17	サウンド注釈
OT_ANT_MOVIE	18	ムービー注釈
OT_ANT_WIDGET	19	ウィジェット注釈
OT_ANT_SCREEN	20	スクリーン注釈
OT_ANT_PRINTERMARK	21	プリンターマーク注釈
OT_ANT_TRAPNET	22	トラップネット注釈
OT_ANT_WATERMARK	23	ウォーターマーク注釈
OT_ANT_3D	24	3D 注釈
OT_ANT_PROJECTION	25	プロジェクション注釈
OT_ANT_RICHMEDIA	26	リッチメディア注釈
OT_ANT_REDACT	27	Redact 注釈

#### AvsObjectState

AvsObjectState	値	説明
OST_SELECTED	0x0001	選択されているか？（取得・設定）
OST_MODIFIED	0x0002	編集状態か？（取得のみ）
OST_OPENED	0x0004	ポップアップがオープンしているか？（取得・設定）
OST_LOCKED	0x0008	編集不可（ロック）であるか？（取得・設定） ※注釈のプロパティダイアログで「ロック」をチェックすること同等です。
OST_NOTSELECTABLE	0x0010	選択不可であるか？（取得・設定） ※選択不可を指定した場合、選択状態は解除されます。また、選択（OST_SELECTED）を指定しても選択状態になりません。

解説：

注釈をプログラムから作成します。オブジェクトパラメータのうち、Coordinates は必ず指定が必要ですので下記表に従って必要な配列をつくり、指定してください。

作成できる注釈は、ASM\_INS\_STICKYNODE ~ ASM\_INS\_REDACT に対応した注釈になります。

※.NET の場合は System.Drawing.Point の配列を ArrayList で指定します。

#### AvsSelectMode

AvsSelectMode	対応する注釈タイプ (AvsObjectType)	Coordinates に指定する必要がある座標
ASM_INS_STICKYNODE	OT_ANT_TEXT	0:左上
ASM_INS_LINK	OT_ANT_LINK	0:左上 1:右下
ASM_INS_FILEATTACHMENT	OT_ANT_FILEATTACHMENT	0:左上
ASM_INS_TEXTBOX	OT_ANT_FREETEXT	0:左上 1:右下
ASM_INS_CALLOUT	OT_ANT_FREETEXT	0:左上 1:右下 2-4: 引出線 (2:矢印-4:テキストボックス接続点)
ASM_INS_TYPEWRITER	OT_ANT_FREETEXT	0:左上 1:右下
ASM_INS_LINE ASM_INS_LINE_DIMENSION	OT_ANT_LINE	0:始点 1:終点
ASM_INS_ARROW	OT_ANT_LINE	
ASM_INS_RECTANGLE	OT_ANT_SQUARE	0:左上 1:右下
ASM_INS_OVAL	OT_ANT_CIRCLE	
ASM_INS_POLYGON ASM_INS_POLYGON_DIMENSION	OT_ANT_POLYGON	点の数だけ順番に指定してください。
ASM_INS_POLYLINE ASM_INS_POLYLINE_DIMENSION	OT_ANT_POLYLINE	
ASM_INS_HIGHLIGHT	OT_ANT_HIGHLIGHT	0:左上 1:右上 2:左下 3:右下
ASM_INS_UNDERLINE	OT_ANT_UNDERLINE	
ASM_INS_CROSSOUT	OT_ANT_STRIKEOUT	
ASM_INS_SQUIGGLY	OT_ANT_SQUIGGLY	
-	OT_ANT_POPUP	0:左上 1:右下
ASM_INS_STAMP_APPROVED~ ASM_INS_STAMP_VOID	OT_ANT_STAMP	0:左上 1:右下
ASM_INS_STAMP_CUSTOM	OT_ANT_STAMP	0:中心点 (中心点指定の場合)

		0:左上 1:右下 (矩形指定の場合) ※矩形指定の場合、縦横比を保持して指定した矩形に収まるように調整されます。
ASM_INS_REDACT	OT_ANT_REDACT	0:左上 1:右下

その他、下記に注意してください。

- Coordinates の座標は指定したページのページサイズに収まるよう指定してください。座標に問題がある場合、位置を補正しますがそれでも問題がある場合はメソッドが失敗します。
- ポップアップ注釈(OT\_ANT\_POPUP)では GetObjectInfo/ModifyObjects でのみ、Coordinates プロパティが利用できます。作成/削除はできません。
- ファイル添付注釈をつくる場合は mask に AOPM\_FILE の指定が必要です。
- AOPM\_FILE の指定をした場合は FileName と FileBin の指定が必須となります。
- カスタムスタンプ注釈をつくる場合は mask に AOPM\_FILE の指定が必要です。  
FileBin の指定が必須となります。

#### 5.5.108. GetObjectCount メソッド

---

**int32 GetObjectCount (int32 pageNo); ([.NET](#))**

戻り値：注釈数

pageNo : ページ番号

解説：指定したページに含まれるオブジェクト（注釈）数を取得します。未対応の注釈はカウントされません。

#### 5.5.109. GetObjectInfo メソッド

---

**AvsDotNetGuiCtl.AvsObject GetObjectInfo(String id, long mask); ([.NET](#))**

戻り値： ([.NET](#)) 成功した場合は取得したオブジェクト情報、失敗した場合は null が返ります。

id : オブジェクト ID

mask : どのパラメータを取得したいかを [AvsObjectMask](#) マスクビットの組み合わせで指定します。

解説：ID を指定してオブジェクト（注釈）情報を取得します。

#### 5.5.110. DeleteObjects メソッド

---

**bool DeleteObjects(ArrayList ids); ([.NET](#))**

戻り値：成功した場合 true が返ります。

ids : オブジェクト ID の配列

解説：指定した ID のオブジェクト（注釈）をすべて削除します。

#### 5.5.111. ModifyObjects メソッド

---

**bool ModifyObjects(ArrayList ids, long mask, AvsDotNetGuiCtl.AvsObject object); (.NET)**

戻り値：成功した場合 true が返ります。

ids : オブジェクト ID の配列

mask : どのパラメータを変更したいかを [AvsObjectMask](#) マスクビットの組み合わせで指定します。

object : オブジェクトのパラメータを指定します。 [AvsObject](#) 参照。

解説：指定した ID のオブジェクト（注釈）をすべて変更します。詳しくは [CreateObject](#) メソッドの説明を参照してください。

#### 5.5.112. HitTest メソッド

---

**AvsDotNetGuiCtl.AvsHitTestResult HitTest (System.Drawing.Point point, [out] String objId); (.NET)**

戻り値：下記結果が返ります。

AvsHitTestResult	値	説明
GHR_NONE	0	Hit せず
GHR_TEXT	1	テキスト
GHR_ANCHOR	2	リンク
GHR_TEXTV	3	テキスト（縦書き）
GHR_OBJ_TOPLEFT	4	注釈 左上
GHR_OBJ_BOTTOMLEFT	5	注釈 左下
GHR_OBJ_BOTTOMRIGHT	6	注釈 右下
GHR_OBJ_TOPRIGHT	7	注釈 右上
GHR_OBJ_TOP	8	注釈 上
GHR_OBJ_LEFT	9	注釈 左
GHR_OBJ_BOTTOM	10	注釈 下

GHR_OBJ_RIGHT	11	注釈 右
GHR_OBJ_INBODY	12	注釈の中
GHR_OBJ_INTEXT	13	注釈 テキストボックスの中
GHR_OBJ_BTN	14	ポップアップ注釈のボタン
GHR_OBJ_INBODY_PARA	15	テキストボックスの境界線
GHR_BADPOINT	1000	用紙外
GHR_OBJ_POINT	10000 以上	多角形／折れ線注釈の点

point [\(.NET\)](#) : クライアント座標

objId : オブジェクトがあった場合にその ID が入ります。

解説：指定した位置にあるものを調べます。オブジェクトが存在する場合はその ID を返します。ただし未対応の注釈は検出できません。

#### 5.5.113. GetNextObjectId メソッド

---

**String GetNextObjectId (int32 pageNo String objId); [\(.NET\)](#)**

戻り値：オブジェクト ID

pageNo : ページ番号

objId : オブジェクト ID

解説：指定したページに存在するオブジェクトの ID を取得します。objId に空文字列を指定すると一つ目のオブジェクト ID が取得でき、ID を指定するとその次の ID が取得できます。

#### 5.5.114. SetHilightText メソッド

---

**void SetHilightText (int32 pageNo, int32 startPos, int32 endPos); [\(.NET\)](#)**

pageNo : ページ番号

startPos : 開始文字位置

endPos : 終了文字位置

解説：指定したページの開始文字位置から終了文字位置までのテキストをハイライト表示します。ハイライト色は HilightTextColor プロパティで変更できます。テキスト選択、検索時のハイライトはこのメソッドで指定されるハイライトより上位に表示されます。

ページサイズの外に位置するテキストは指定してもハイライトされません。

#### 5.5.115. ClearHilightText メソッド

---

**void ClearHilightText();** [\(.NET\)](#)

解説：SetHilightText メソッドで設定されたハイライトをすべて解除します。

#### 5.5.116. GetTextInfo メソッド

---

**ArrayList GetTextInfo (int32 pageNo, AvsTextExtractOption option);** [\(.NET\)](#)

戻り値：AvsTextInfo オブジェクトの配列

pageNo : ページ番号

option : テキストの抽出条件

解説：指定したページの矩形範囲に含まれるテキストについて、テキストの情報（AvsTextInfo オブジェクト）を文字ごとに配列で返します。矩形範囲は AvsExtractTextOption の Rects フィールドで指定してください。

TEXT\_SORT を指定した場合、テキストの座標に従ってソートされた順でテキストが抽出されます。また文字間や行間が認識された場合は、ATT\_Space や ATT\_Newline のテキスト情報が含まれる場合があります。通常のテキスト情報（ATT\_Normal）で Text に空白や改行文字が含まれる場合があります。

TEXT\_RAW を指定した場合、ページコンテンツの順番でテキストが抽出されます。この場合はテキストの座標値は利用できません。また ATT\_Space や ATT\_Newline のテキスト情報は追加されません。

※座標変換により文字に回転や歪み（skew）がある場合など正しい情報が取得できない場合があります。

※縦書きや斜めに傾いている場合には正しく機能しません。

※取得できる情報やテキストの順序は GetTextInRect とは同一にはなりません。

AvsDotNetGuiCtl.AvsTextInfo メンバ		
型	メンバ名	説明
.NET		
AvsTextType	TextType	

String	Text	文字
System.Drawing.Rectangle	Rect	文字のバウンディングボックス（論理座標での原点からの位置）
int32	Baseline	ベースライン位置（論理座標での原点からの高さ）単位 TWIP
int32	FontSize	フォントサイズ 単位 TWIP
String	FontName	フォント名（ベースフォント名）
Int32	PageNumber	文字が存在するページ番号
AvsWritingMode	WritingMode	文字列の方向を表す数値です。方向は現在の座標系での方向になります。（ViewCoord プロパティに依存します。） ※座標変換に 0/90/180/270 度以外の回転がある場合は文字列の方向が正しく取得できません。
bool	IsVertical	PDF 内部のフォントが縦書き用か横書き用か  true … 縦書き  false … 横書き  ※ページ回転や座標変換によりません

#### プロパティ

IsNewLine 型： bool (.NET) (getのみ) テキスト情報が空白か。

IsSpace 型： bool (.NET) (getのみ) テキスト情報が改行か。

#### 取得可能なフィールド

AvsTextInfo クラスの取得可能なフィールドは、取得するメソッドによって変わります。

取得メソッド	GetTextInfo (TEXT_RAW)	GetTextInfo (TEXT_SORT)	GetTextInfoInRect /GetSelectedTextInfo
Text	○	○	○
Rect	×	○※3	○※3
Baseline	×	×	○
FontSize	×	○※1	○
FontName	×	×	○
PageNumber	○	○	○
WritingMode	×	×	○
IsVertical	×	×	○※2

- ※1 同じフォントであっても、座標変換の拡大縮小によって出現箇所によって実際の表示サイズはことなります。この拡大縮小が反映された見かけ上のサイズになります。
- ※2 IsNewLine フィールドは V5.0 改訂 6 版から廃止されました。代わりに IsNewLine プロパティ (get) をお使いください。
- ※3 Rect フィールドのサイズは、GetTextInfo と GetTextInfoInRect では必ずしも一致しません。

#### AvsTextType

AvsTextType	値	説明
ATT_Normal	0	文字（改行や空白文字が含まれる場合もあります）
ATT_Newline	1	行間から認識された改行
ATT_Space	2	文字間から認識された空白

#### AvsWritingMode

AvsWritingMode	値	説明
WM_Ir_tb	0	左から右
WM_rl_tb	1	右から左
WM_tb_rl	2	上から下
WM_bt_rl	3	下から上
WM_Ir_bt	4	左から右（WM_Ir_tb の上下反転）
WM_rl_bt	5	右から左（WM_rl_tb の上下反転）
WM_tb_lr	6	上から下（WM_tb_rl の左右反転）
WM_bt_lr	7	下から上（WM_bt_rl の左右反転）

#### AvsTextExtractOption

AvsDotNetGuiCtl.AvsTextExtractOption メンバ		
型	メンバ名	説明
.NET		
AvsTextExtractType	TextType	TEXT_RAW または TEXT_SORT
float	IgnoreActualText	ActualText を無視するか 既定値 : false (無視しない)

float	TextOverlapRatio	テキストが矩形とどれくらい重なっていたら抽出対象とするか 既定値：0.3
float	OverlapAsLine	文字と文字がどれくらい重なっていると同じ行とみなすか (TEXT_SORTのみ) 既定値：0.7
float	SameTextOmitRatio	同じ文字が重なっている場合、文字と文字がどれだけ重なっていると取り除くか (TEXT_SORTのみ) 既定値：0.0 (取り除かない)
float	DifferentTextOmitRatio	違う文字が重なっている場合、文字と文字がどれだけ重なっていると取り除くか (TEXT_SORTのみ) 既定値：0.0 (取り除かない)
bool	DifferentTextOmitRatioApplyAtDifferentColor	DifferentTextOmitRatio を色が違う場合にのみ適用するか (TEXT_SORTのみ) 既定値：true
Int32	UnicodeToSpace	指定された Unicode を Space(U+0020)に置き換えるか 既定値：0 (置き換えない)
Int32	UnicodeToRemove	指定された Unicode を Space(U+0020)に削除するか 既定値：0 (削除しない)
ArrayList	Rects	System.Drawing.Rectangle を要素にもつ配列。テキスト抽出の対象とする矩形領域を指定してください。複数の領域を指定できます。 ページ全体の場合は空配列を指定してください。

#### AvsTextExtractType

AvsTextExtractType	値	説明
TEXT_RAW	1	ページコンテンツにおける出現順で抽出する
TEXT_SORT	2	取得したテキストを座標でソートして抽出する

#### サンプルコード例

- GetTextInfoInRect は以下の対応するコードでおきかえ可能です。

```
//ArrayList list = pdfPageViewCtrl1.GetTextInfoInRect(pageNo, rc);

AvsTextExtractOption opt = new AvsTextExtractOption();

opt.TextType = AvsTextExtractType.TEXT_RAW;

opt.Rects.Add(rc);

ArrayList list = pdfPageViewCtrl1.GetTextInfo(pageNo, opt);
```

- 余白から判定された文字間や行間のマークは以下で判定できます。

```
AvsTextInfo ti = (AvsTextInfo)list[i];

if (ti.Is.NewLine) { } // 余白から判定された行間

else if (ti.IsSpace) { } // 余白から判定された文字間

else { } // 文字コードが存在する場合。空白スペース等も含む
```

### 5.5.117. GetTextInfoInRect メソッド

---

**ArrayList GetTextInfoInRect (int32 pageNo, System.Drawing.Rectangle rect); (.NET)**

戻り値 : AvsTextInfo オブジェクトの配列

pageNo : ページ番号

rect : 矩形範囲

解説 : 指定したページの矩形範囲に含まれるテキストについて、テキストの情報 (AvsTextInfo オブジェクト) を文字ごとに配列で返します。

テキストが改行される場合は、その位置に IsNewLine=True となる、AvsTextInfo オブジェクトが追加されます。

※座標変換により文字に回転や歪み (skew) がある場合など正しい情報が取得できない場合があります。※現在の実装では、テキストのベースラインに一定量変化があるときに改行と判定しています。縦書きや斜めに傾いている場合には正しく機能しません。

※改行位置の判定は PDF 仕様では明確に既定されないため、弊社リーダーの実装依存の動作になります。このため、他社のリーダーと同一にはなりません。

※取得できる情報やテキストの順序は GetTextInfo とは同一にはなりません。

### 5.5.118. ClientToLogicalRect メソッド

---

**System.Drawing.Rectangle ClientToLogicalRect (int32 pageNo System.Drawing.Rectangle rect); (.NET)**

戻り値 (.NET) : 矩形 (論理座標)

pageNo (.NET) : ページ番号

rect ([.NET](#)) : 矩形（クライアント座標）

解説：クライアント座標で表現された矩形を指定したページの論理座標に変換します。

#### 5.5.119. SetCtrlCursor メソッド

---

**System.Windows.Forms.Cursor SetCtrlCursor (System.Windows.Forms.Cursor cursor); ([.NET](#))**

戻り値：カーソルオブジェクト または NULL

cursor : カーソルオブジェクト

解説：コントロールのカーソルを指定したカーソルに指定します。通常はコントロール自身がカーソルを状態に合わせて制御するため、CtrlStyle プロパティの AGCS\_DISABLECURSOR フラグを ON に設定して、コントロールのカーソル制御を無効にしてください。

なお、カーソルが変更されるのは PDFPageViewCtrl にフォーカスしている最中のみです。上位アプリケーションにフォーカスがある場合はカーソルは変更されません。別途、Cursor.Current プロパティなどで変更する必要があります。

#### 5.5.120. SetSmoothing メソッド

---

**bool SetSmoothing (bool text, bool lineart, bool image,); ([.NET](#))**

戻り値：再描画が必要な場合 true を返します。

text : 文字のスムージング(AntiAlias)を行います。

lineart : 線画のスムージングを行います。

image : イメージのスムージングを行います。

解説：スムージングを設定します。初期値はすべて true です。

変更はいつでも行えます。設定変更後、必要な場合は再描画されます。

表示だけでなく、印刷、イメージ作成にも有効です。

#### 5.5.121. SelectTextString メソッド

---

**void SelectTextString (LogicalPoint start, LogicalPoint end); ([.NET](#))**

start : 選択の開始位置（論理座標）

end : 選択の終了位置（論理座標）

解説：指定したテキストの範囲を選択状態（ハイライト）にします。

開始位置・終了位置のどちらかに文字がない場合は、選択状態になりません。

ClearSelect メソッドで選択状態を解除できます。

#### 5.5.122. SetFocus メソッド

---

**void SetFocus(void);** [\(.NET\)](#)

解説：コントロールにフォーカスを設定します。.NET の Control.Focus メソッドは正しく機能しませんのでこちらを使用してください。

#### 5.5.123. GetSelectedTextInfo メソッド

---

**void GetSelectedTextInfo(void);** [\(.NET\)](#)

解説：選択中のテキストについて、テキストの情報（AvsTextInfo オブジェクト）を文字ごとに配列で返します。  
AvsTextInfo および制限事項について、[GetTextInfoInRect メソッド](#) を参照ください。

#### 5.5.124. FitViewToRect メソッド

---

**void FitViewToRect(AvsDotNetGuiCtl.LogicalPoint /Pos1, AvsDotNetGuiCtl.LogicalPoint /Pos2);** [\(.NET\)](#)

IPos1 : 左上の論理座標 [\(.NET\)](#)

IPos2 : 右下の論理座標 [\(.NET\)](#)

解説：二点で指定される矩形範囲を表示します。縦横比を維持して指定した矩形がコントロールウィンドウに収まる倍率で表示されます。

座標値は ViewCoord の設定にあわせて論理座標またはビュー座標を指定してください。異なる  
PageNumber を持つ LogicalPoint を指定できますが PageLayout にあった座標を指定してください。

以下の制限があります。

- ・最大の倍率（6400%）以上になる範囲は指定しても最大倍率でしか表示されません。
- ・コントロールサイズやスクロールバーの関係で表示範囲はおおよそ指定された範囲が表示されますが、指定した座標値にぴったり一致はしません。

- 現在の PageLayout で表示できない範囲の場合、意図しない表示結果になる場合があります。PageLayout は自動で調整されません。

### 5.5.125. GetLicenseInfo メソッド

---

**string GetLicenseInfo(AvsLicenseInfo type); (.NET)**

type : 取得するライセンス情報の種別

戻り値(.NET) : ライセンス情報を表す文字列

設定値 :

AvsLicenseInfo (enum 値)	値	説明
LI_PATH	0	ライセンスファイル (pdfviewersdk.lic) の絶対パス
LI_SERIAL	1	シリアルナンバー
LI_COMPANY	2	会社名
LI_SECTION	3	部署名
LI_USERNAME	4	ユーザー名
LI_MAINTENANCE_LIMIT	5	保守期限

解説 : 指定されたライセンス情報を取得します。

ライセンスファイルに登録がない場合、空文字列が返る場合があります。ライセンスファイルのロードに失敗した場合、LI\_PATH の戻り値としてエラーメッセージを返します。

ライセンスファイルのロードに失敗した場合、LI\_PATH の戻り値としてエラーメッセージを返します。

### 5.5.126. ShowMeasureDialog メソッド

---

**void ShowMeasureDialog(); (.NET)**

解説 : 測定比率と精度ダイアログを表示します。この設定にしたがって距離ツール／周辺ツール／面積ツールの測定値が計算されます。左側がページ上の単位、右側がものさしツールの測定値の単位です。

項目	解説
単位 (変換元)	pt(ポイント)、in (インチ)、cm (センチメートル)、mm (ミリメートル)、pi (パイカ)
単位 (変換先)	変換元の単位に加え以下の単位。 ft (フィート)、yd (ヤード)、m (メートル)、km (キロメートル)、mi (マイル)

精度	1、0.1、0.01、0.001、0.0001
----	-------------------------

#### 制限事項

- 現在、測定比率と精度の既定値を保存する機能はありません。

#### 5.5.127. RedactAndSaveDocument メソッド

---

**int32 RedactAndSaveDocument(string filePath); (.NET)**

filePath : PDF ファイルのパスを指定します。

戻り値：成功した場合 0 が返ります。失敗した場合はエラーコードが返ります。

解説： PDF ファイルを保存します。

墨消し (Redact 注釈) を反映して指定されたパスに PDF ファイルを保存します。墨消しの反映は以下のように行われます。

- Redact 注釈の範囲に含まれるテキストは文字単位で削除されます。
- 画像/パス/シェーディングは完全に矩形に含まれている場合のみ削除されます。
- 削除されたテキスト等のオブジェクトは、ページコンテンツから完全に削除されます。保存後の PDF からは復元できません。
- Redact 注釈の範囲のページコンテンツはマスクカラーで塗りつぶされます。
- 元の Redact 注釈はすべて削除されます。

保存されていない編集がある場合は失敗します。予め PDF ファイルを保存してから実行してください。

エラー時にダイアログを一切表示したくない場合は CtrlStyle プロパティで 0x0004 を指定してください。

(制限事項)

- 不正な PDF ファイルである場合や不具合により、正しく墨消しがされない、保存後に元の PDF の内容が変わ、PDF ファイルが破損するなど、正しく処理されない場合があります。元のファイルを上書きしないようご注意ください。

#### 5.5.128. DocumentOpened イベント

---

**void DocumentOpened(IntPtr hPDFDocument); (.NET)**

hPDFDocument : PDFDocument のハンドル

解説：ドキュメントがオープンされたときに発行されます。

### 5.5.129. PageTurned イベント

---

**void PageTurned(int32 pageNo);** [\(.NET\)](#)

pageNo : ページ番号

解説：現在のページ番号が変更になったときに発行されます。

### 5.5.130. Zoom イベント

---

**void Zoom(float32 ratio);** [\(.NET\)](#)

ratio : ズーム倍率

解説：ズーム倍率が変更になったときに発行されます。

### 5.5.131. DragDropFiles イベント

---

**int32 DragDropFiles(BSTR[] files);** [\(.NET\)](#)

files : ドロップされたファイル名

戻り値[\(.NET\)](#) : デフォルトの動作 (files[0]を開く) を実行する場合は 0 以外を返します。 0 を返すとデフォルト動作をおこないません。

解説：コントロールにファイルがドラッグ & ドロップされたときに発行されます。

### 5.5.132. ObjectAdded イベント

---

**void ObjectAdded(string objectId);** [\(.NET\)](#)

objectId : 追加されたオブジェクトの UUID

解説：オブジェクト挿入直後に発行されます。必要ならここで選択モードを元に戻すなどします。

### 5.5.133. DoAction イベント

---

**int32 DoAction(AvsDotNetGuiCtl.AvsActionType actionType, string data);** [\(.NET\)](#)

戻り値 [\(.NET\)](#) : アクションを実行する場合は 1 を返します。 0 を返すとアクションの実行を中止します。

actionType : アクションの種類

AvsActionType	値	説明
AGAT_GOTO	1	現在のドキュメント
AGAT_GOTOR	2	他のドキュメント
AGAT_LAUNCH	4	他のファイルを開く
AGAT_URI	6	Web ブラウザで URL を開く
AGAT_INTERNALFILE	1000	添付ファイル注釈のファイルを開く

data : 表示しようとしているファイル名、URL が入ります。

解説：アクション実行時、添付ファイル注釈のファイルを開くときに発行されます。ファイルを開くときに警告メッセージを表示するためなどに使用します。

#### 5.5.134. DefaultObjectPropertyChanged イベント

**void DefaultObjectPropertyChanged(AvsDotNetGuiCtl.AvsSelectMode type);** ([.NET](#))

type : 設定するオブジェクトタイプ。ASM\_INS\_STICKYNODE～ASM\_INS\_SQUIGGLY の間で指定します。※  
[SetDefaultObjectProperty](#) 参照

解説：オブジェクトの右クリックメニューから「現在のプロパティをデフォルトとして使用」が選ばれデフォルトのオブジェクトプロパティが変更されたときに発行されます。

デフォルトプロパティをファイル等に保存する処理をおこないたい場合にこのイベントをハンドリングしてください。

#### 5.5.135. MouseDownC イベント

**int32 MouseDownC(AvsDotNetGuiCtl.AvsMouseBtnType btnType, long nFlags,**  
**System.Drawing.Point point);** ([.NET](#))

戻り値 ([.NET](#)) : コントロールのデフォルト処理を実行する場合は 1 を返します。 0 を返すとデフォルト処理をおこないません。

btnType : マウスボタンの種類

AvsMouseBtnType	値	説明
AMBT_LBUTTON	0	左ボタン
AMBT_RBUTTON	1	右ボタン
AMBT_MBUTTON	2	中央ボタン

nFlags : 未使用です。

point (.NET) : カーソル位置のクライアント座標が入ります。

解説：マウスボタンが押されたときに発行されます。

#### 5.5.136. MouseUpC イベント

---

```
int32 MouseUpC(AvsDotNetGuiCtl.AvsMouseBtnType btnType, long nFlags,  
                  System.Drawing.Point point); (.NET)
```

戻り値 (.NET) : コントロールのデフォルト処理を実行する場合は 1 を返します。 0 を返すとデフォルト処理をおこないません。

btnType : マウスボタンの種類 [AvsMouseBtnType](#) 参照。

nFlags : 未使用です。

point (.NET) : カーソル位置のクライアント座標が入ります。

解説：マウスボタンが離されたときに発行されます。

#### 5.5.137. DoubleClickC イベント

---

```
int32 DoubleClickC(AvsDotNetGuiCtl.AvsMouseBtnType btnType, long nFlags,  
                     System.Drawing.Point point); (.NET)
```

戻り値 (.NET) : コントロールのデフォルト処理を実行する場合は 1 を返します。 0 を返すとデフォルト処理をおこないません。

btnType : マウスボタンの種類 [AvsMouseBtnType](#) 参照。

nFlags : 未使用です。

point (.NET) : カーソル位置のクライアント座標が入ります。

解説：ダブルクリックされたときに発行されます。

#### 5.5.138. MouseMoveC イベント

---

```
int32 MouseMoveC(long nFlags, System.Drawing.Point point); (.NET)
```

戻り値 (.NET) : コントロールのデフォルト処理を実行する場合は 1 を返します。 0 を返すとデフォルト処理をおこないません。

nFlags : 未使用です。

point (.NET) : カーソル位置のクライアント座標が入ります。

解説：マウスカーソルが移動したときに発行されます。

#### 5.5.139. ContextMenuC イベント

---

**int32 ContextMenuC(System.Drawing.Point point); (.NET)**

戻り値 (.NET) : コントロールのデフォルト処理を実行する場合は 1 を返します。 0 を返すとデフォルト処理をおこないません。

point (.NET) : カーソル位置のクライアント座標が入ります。

解説：マウス右クリックされるかコンテキストメニューボタンが押されたときに発行されます。

#### 5.5.140. KeyDownC イベント

---

**int32 KeyDownC(long nChar); (.NET)**

戻り値 (.NET) : コントロールのデフォルト処理を実行する場合は 1 を返します。 0 を返すとデフォルト処理をおこないません。

nChar : 押されたキーの仮想キーコードです。

解説：キーが押されたときに発行されます。

#### 5.5.141. KeyUpC イベント

---

**int32 KeyUpC(long nChar); (.NET)**

戻り値 (.NET) : コントロールのデフォルト処理を実行する場合は 1 を返します。 0 を返すとデフォルト処理をおこないません。

nChar : 離されたキーの仮想キーコードです。

解説：キーが離されたときに発行されます。

#### 5.5.142. ObjectDeleted イベント

---

**void ObjectDeleted(string objectId);** [\(.NET\)](#)

objectId : 削除されたオブジェクトの UUID

解説：オブジェクト削除直後に発行されます。この時点ではオブジェクトは存在しないので、objectId をキーにしてオブジェクトの情報を取得する事はできません。

#### 5.5.143. ObjectResized イベント

---

**void ObjectResized(string objectId);** [\(.NET\)](#)

objectId : リサイズされたオブジェクトの UUID

解説：オブジェクトがリサイズされると発行されます。引き出し線注釈の矢印部分や、多角形注釈の点など端点の移動でも発行されます。

#### 5.5.144. ObjectMoved イベント

---

**void ObjectMoved(string objectId);** [\(.NET\)](#)

objectId : 移動されたオブジェクトの UUID

解説：オブジェクトが移動されると発行されます。

#### 5.5.145. EditBoxShowed イベント

---

**void EditBoxShowed(bool showed, string objectId);** [\(.NET\)](#)

showed : true テキスト入力ボックス表示、 false テキスト入力ボックス非表示

objectId : オブジェクトの UUID

解説：テキスト入力ボックスが表示／非表示となると発行されます。objectId はテキスト入力が開始または終了されたテキストボックス等の注釈の ID です。

#### 5.5.146. EditBoxTextEdited イベント

---

**void EditBoxTextEdited(string objectId);** [\(.NET\)](#)

objectId : オブジェクトの UUID

解説：テキスト入力ボックスのテキストが編集されると発行されます。

#### 5.5.147. Scrolled イベント

---

**void Scrolled();** [\(.NET\)](#)

解説：水平または垂直スクロールバーによって表示がスクロールされると発行されます。

## 6. PDFThumbnailCtrl

---

### 6.1. 概要

サムネイルを表示するコントロールです。フォームに貼り付けるだけで簡単に利用することができます。

### 6.2. チュートリアル .NET Framework 版

以下では VisualStudio C#での初步的な使用方法を説明します。.NET Framework 版を利用するには事前に.NET Framework 4.6.2 以降のインストールが必要です。 詳細は「4.2 動作に必要なモジュール、手続き等」をご覧ください。以下は 32bit 版での説明です。

- ① VisualStudio を起動する前に環境変数"Path"に bin¥Win32 フォルダのパスを追加してください。こうしないとフォームデザイナを開いたときエラーになることがあります。（ビルド、実行には影響しませんので開発環境のみおこなってください。64bit 版コントロールを利用した開発については「64bit コントロールについて」を参照してください。）
- ② メニューから[ファイル]-[新規作成]-[プロジェクト]を選び、「新しいプロジェクト」ダイアログから VisualC# の「Windows フォーム アプリケーション」を選択して適当なプロジェクト名（ここでは"Test"とします）を入力したら[OK]を押してください。
- ③ メニューから[ツール]-[ツールボックス アイテムの選択]を選び、右下の「参照...」ボタンを押して bin¥Win32 ディレクトリにある AvsDotNetGuiCtl.dll を選択してください。「PDFThumbnailControl」にチェックをつけて[OK]を押します。
- ④ ツールボックススペインに「PDFThumbnailControl」のアイコンが追加されていますのでそれを選びフォーム上でドラッグ＆ドロップして貼り付けてください。その他、 OpenFileDialog コントロール、Button コントロールを貼り付けてください。
- ⑤ 貼り付けた Button コントロールをダブルクリックします。コードが表示されますので次のように処理を書きます。

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
        pdfThumbnailCtrl1.OpenDocument(openFileDialog1.FileName, "");
}
```

- ⑥ メニューから[プロジェクト]-[参照の追加]を選び[参照]タブを選択し、bin¥Win32 ディレクトリにある AvsDotNetGuiCtl.dll を選択してください。[OK]を押すとプロジェクトの参照設定に AvsDotNetGuiCtl が追加されます。
- ⑦ 実行します。ボタンを押すとファイルダイアログが表示され PDF ファイルを指定するとサムネイルを表示することができます。

### 6.3. アプリケーションが実装すべき項目

コントロールには多くの機能がデフォルトで実装されており、ほとんどの機能はボタンを用意してメソッドを呼び出すだけで、もしくはファイルを開いただけで、簡単に利用できるようになっています。ここでは各機能についてアプリケーション側が実装すべきこと、関知できることについて簡単に説明します。詳しくは付属のサンプルプログラムを参考にしてください。

機能概要	説明
ファイルのドラッグ & ドロップ	デフォルトではドロップされたファイルの一つを開きます。 異なる動作をおこないたい場合は DragDropFiles イベントをハンドリングしてください。
コントロール間の同期（編集）	デフォルトではなにもしません。 同期をとるには DocumentOpened イベントで AttachPDFDocument してください。同期をとるとページの入れ替えや回転といった編集処理をおこなったとき、同期中の全コントロールに反映されます。
コントロール間の同期（ページ遷移）	デフォルトではなにもしません。 同期をとるには PageTurned イベントで GotoPage してください。
右クリックメニュー	デフォルトで用意されています。 これを任意のものに変更することはできません。
ページのドラッグ & ドロップ	ページのドラッグ & ドロップによる移動、コピー処理がデフォルトで用意されています。 同ドキュメント内では [Ctrl] を押しながらドロップするとコピーになります。また、異なるドキュメントからドラッグ & ドロップでページをもってくることが可能です。
ページの移動、コピー、削除	ページを処理するための GUI インタフェースはデフォルトで用意されています。 ショートカットキー [Delete]/[BackSpace]/[Esc]/[Ctrl]/[Shift]/方向キーがデフォルトでハンドリングされています。意図的な別処理への置き換えでなければアプリケーション側ではハンドリングしないでください。
アンドゥー、リドゥー	デフォルトで実装されています。 独自のアンドゥー処理を差し込むことはできません。 ショートカットキー [Ctrl]+[Z]/[Y] がデフォルトでハンドリングされていますので意図的に別の処理に置き換えたいためなければアプリケーション側ではハンドリングしないでください。
スクロール	スクロールバークリック、キー押下などによるスクロール処理がデフォルトで実装されています。

	ショートカットキー[Home]/[End]/[PageUp]/[PageDown]/方向キーがデフォルトでハンドリングされています。意図的な別処理への置き換えでなければアプリケーション側ではハンドリングしないでください。
カット、コピー & ペースト、すべて選択	デフォルトで実装されています。 ショートカットキー[Ctrl]+[X]/[C]/[V]/[A]がデフォルトでハンドリングされています。意図的な別処理への置き換えでなければアプリケーション側ではハンドリングしないでください。

## 6.4. API 一覧

### 6.4.1. プロパティ

プロパティ名	.NET / .NET Framework	概要
IsOpen	○	PDF がオープンされているか調べます。
IsModified	○	PDF が編集状態か調べます。
IsLockedByOwnerPassword	○	オーナーパスワードでロックされている状態かどうか調べます。
HasFocus	○	コントロールがフォーカスを持っているかどうか調べます。
CurrentPageNumber	○	現在のページ番号を取得します。
PageCount	○	ページ数を取得します。
SelectedPageCount	○	選択中のページ数を取得します。
CtrlStyle	○	スタイルを設定します。
BackgroundColor	○	背景色を設定します。
TextColor	○	テキスト色を設定します。
ThumbnailSize	○	サムネイルサイズを設定します。
CanUndo	○	Undo が可能か調べます。
CanRedo	○	Redo が可能か調べます。
CanCut	○	Cut が可能な状態か調べます。
CanCopy	○	Copy が可能な状態か調べます。
CanPaste	○	Paste が可能な状態か調べます。
CanDelete	○	Delete が可能な状態か調べます。
CanModifyPage	○	ページ編集が可能な状態か調べます。
HasUserPassword	○	ユーザーパスワードが設定されているかどうかを調べます。
HasOwnerPassword	○	オーナーパスワードが設定されているかどうかを調べます。
UserPassword	○	ユーザーパスワードを取得/設定します。
OwnerPassword	○	オーナーパスワードを取得/設定します。
SecurityPermissontFlag	○	PDF のセキュリティフラグを取得/設定します。
DocumentPath	○	オープンしている PDF の Path を取得します。
SDKVersion	○	SDK バージョン番号を取得します。

SelectedPages	<input type="radio"/>	選択中のページ番号のリストを返します。
---------------	-----------------------	---------------------

#### 6.4.2. メソッド

---

メソッド名	.NET / .NET Framework	概要
OpenDocument	<input type="radio"/>	PDF をオープンします。
SaveDocument	<input type="radio"/>	PDF を保存します。
CloseDocument	<input type="radio"/>	PDF をクローズします。
GetPDFDocumentHandle	<input type="radio"/>	PDFDocument のハンドルを取得します。
AttachPDFDocument	<input type="radio"/>	PDFDocument のハンドルをコントロールに関連付けます。
UnlockOwnerPasswordSecurity	<input type="radio"/>	オーナーパスワードを指定してセキュリティを解除します。
GotoPage	<input type="radio"/>	指定したページへ移動します。
GetPageRotation	<input type="radio"/>	ページの回転角度を取得します。
RotatePages	<input type="radio"/>	ページを回転します。
InsertNewPages	<input type="radio"/>	白紙ページを挿入します。
MovePages	<input type="radio"/>	ページを移動します。
CopyPages	<input type="radio"/>	ページをコピーします。
ResetView	<input type="radio"/>	ビューをクリアして作り直します。
Undo	<input type="radio"/>	編集を元に戻します。
Redo	<input type="radio"/>	Undo をやり直します。
Cut	<input type="radio"/>	切り取ってクリップボードに保存します。
Copy	<input type="radio"/>	コピーしてクリップボードに保存します。
Paste	<input type="radio"/>	クリップボードの内容を貼り付けます。
Delete	<input type="radio"/>	削除をおこないます。
SelectPage	<input type="radio"/>	ページを選択します。
SelectAll	<input type="radio"/>	全ページを選択します。

#### 6.4.3. イベント

---

イベント名	.NET / .NET Framework	概要
DocumentOpened	○	ドキュメントがオープンされたときに呼ばれます。
PageTurned	○	選択ページが変わったときに呼ばれます。
DragDropFiles	○	ファイルがドラッグ & ドロップされたときに呼ばれます。
ContextMenuC	○	コントロール上でマウスボタンが右クリックされたときやコンテキストメニューボタンが押されたときに呼ばれます。

## 6.5. API リファレンス

### 6.5.1. IsOpen プロパティ

---

型: bool (.NET) (getのみ)

解説: PDF がオープンされているかどうか調べます。オープンされていれば True になります。

### 6.5.2. IsModified プロパティ

---

型: bool (.NET) (getのみ)

解説: 最後に保存してから PDF が変更されているかどうかを調べます。変更されていれば True になります。

### 6.5.3. IsLockedByOwnerPassword プロパティ

---

型: bool (.NET) (getのみ)

解説: PDF にオーナーパスワードセキュリティによるロックがかかっている状態かどうかを調べます。ロックされていれば True になります。

### 6.5.4. HasFocus プロパティ

---

型: bool (.NET) (getのみ)

解説: コントロールがフォーカスを持っているかどうか調べます。フォーカスをもっていれば True になります。.NET の Control.Focused プロパティは正しく機能しませんのでこのプロパティを使用してください。フォーカスを設定するには SetFocus メソッドを使用してください。 (.NETのみ)

### 6.5.5. CurrentPageNumber プロパティ

---

型: int32 (.NET) (getのみ)

解説：現在のページ番号です。

#### 6.5.6. PageCount プロパティ

---

型：int32 (.NET) (getのみ)

解説：PDF の総ページ数です。PDF がオープンされていない場合は 0 になります。

#### 6.5.7. SelectedPageCount プロパティ

---

型：int32 (.NET) (getのみ)

解説：選択中のページ数が返ります。

#### 6.5.8. CtrlStyle プロパティ

---

型：int32 (.NET) (get / set)

設定値：

AvsCtrlStyle	値	説明
AGCS_HIDEOBJECTS	0x0001	注釈等のオブジェクトを表示しません
AGCS_HIDEERRDLG	0x0004	エラーダイアログを表示しません
AGCS_HIDEPASSWORDDLG	0x0008	パスワード入力ダイアログを表示しません
AGCS_READONLY	0x0010	読み専用モード（編集不可）に設定します

解説：コントロールのスタイルを上記ビットの組み合わせで設定します。デフォルトは 0 です。

#### 6.5.9. BackgroundColor プロパティ

---

型：System.Drawing.Color (.NET) (get / set)

解説：背景色（ページの外の領域）です。

変更する場合は OpenDocument 前に設定してください。OpenDocument 後に設定しても反映されません。

#### 6.5.10. TextColor プロパティ

---

型： System.Drawing.Color (.NET) (get / set)

解説：ページ番号のテキスト色です。

変更する場合は OpenDocument 前に設定してください。OpenDocument 後に設定しても反映されません。

#### 6.5.11. ThumbnailSize プロパティ

---

型： int32 (.NET) (get / set)

解説：表示するサムネイルのサイズです。単位はピクセルです。デフォルトは 82 です。

設定可能な値の範囲は、30～200 です。変更する場合は OpenDocument 前に設定してください。OpenDocument 後に設定しても反映されません。

#### 6.5.12. CanUndo プロパティ

---

型： bool (.NET) (get のみ)

解説：Undo ができるかどうか調べます。Undo 可能であれば True になります。

#### 6.5.13. CanRedo プロパティ

---

型： bool (.NET) (get のみ)

解説：Redo ができるかどうか調べます。Redo 可能であれば True になります。

#### 6.5.14. CanCut プロパティ

---

型： bool (.NET) (get のみ)

解説：Cut ができるかどうか調べます。Cut 可能であれば True になります。

#### 6.5.15. CanCopy プロパティ

---

型： bool (.NET) (getのみ)

解説：Copy ができるかどうか調べます。Copy 可能であれば True になります。

#### 6.5.16. CanPaste プロパティ

---

型： bool (.NET) (getのみ)

解説：Paste ができるかどうか調べます。Paste 可能であれば True になります。

#### 6.5.17. CanDelete プロパティ

---

型： bool (.NET) (getのみ)

解説：Delete ができるかどうか調べます。Delete 可能であれば True になります。

#### 6.5.18. CanModifyPage プロパティ

---

型： bool (.NET) (getのみ)

解説：ページの編集（追加、削除、変更）ができるかどうか調べます。ページ編集可能であれば True になります。

#### 6.5.19. HasUserPassword プロパティ

---

型： bool (.NET) (getのみ)

解説：PDF にユーザパスワードが設定されているか調べます。ユーザパスワードが設定されていれば True になります。

通常はオープン時の PDF の状態を返しますが、UserPassword プロパティに値をセットした場合、状態が変更されます。ドキュメントをオープンしていない場合は必ず False になります。

#### 6.5.20. HasOwnerPassword プロパティ

---

型： bool ([.NET](#)) ([get](#) のみ)

解説：PDF にオーナーパスワードが設定されているか調べます。オーナーパスワードが設定されていれば True になります。

通常はオープン時の PDF の状態を返しますが、OwnerPassword プロパティに値をセットした場合、状態が変更されます。ドキュメントをオープンしていない場合は必ず False になります。

#### 6.5.21. UserPassword プロパティ

---

型： string ([.NET](#)) ([get / set](#))

解説：ユーザーパスワードです。オープン時にユーザーパスワードを指定して PDF を開くとその値が入ります。空文字をセットするとパスワードを設定していないことになり、HasUserPassword で False が返るようになります。

値のセットは IsLockedByOwnerPassword() が True の状態では失敗します。

#### 6.5.22. OwnerPassword プロパティ

---

型： string ([.NET](#)) ([get / set](#))

解説：オーナーパスワードです。オープン時にオーナーパスワードを指定して PDF を開くとその値が入ります。空文字をセットするとパスワードを設定していないことになり、HasOwnerPassword で False が返るようになります。

値のセットは IsLockedByOwnerPassword() が True の状態では失敗します。

#### 6.5.23. SecurityPermissionFlag プロパティ

---

型： int32 ([.NET](#)) ([get / set](#))

解説：PDF のセキュリティ許可フラグです。

次のビットの組み合わせとなります。

AvsSecurityPermissionFlag	値	説明
ASPF_ENABLE_PRINT	0x0004	印刷許可
ASPF_ENABLE_CHANGEDOC	0x0008	ドキュメント内容の変更許可
ASPF_ENABLE_COPYCONTENT	0x0010	内容のコピー許可
ASPF_ENABLE MODIFYANNOT	0x0020	注釈の編集許可
ASPF_ENABLE_FILLFORM	0x0100	フォームフィールドの入力許可
ASPF_ENABLE_ACCESSIBILITY	0x0200	アクセシビリティの許可
ASPF_ENABLE_ASSEMBLEDDOC	0x0400	ページの挿入、削除、回転許可
ASPF_ENABLE_PRINTHIGHLEVEL	0x0804	高解像度の印刷許可

値のセットは IsLockedByOwnerPassword が True の状態では失敗します。

#### 6.5.24. DocumentPath プロパティ

---

型： string (.NET) (getのみ)

解説：オープン中 PDF ファイルの Path を取得します。

V5.0 以降では SaveDocument でファイルパスを指定した後もオープンしたパスを返します。

#### 6.5.25. SDKVersion プロパティ

---

型： string (.NET) (getのみ)

解説：PDFViewerSDK のバージョン番号です。

#### 6.5.26. SelectedPages プロパティ

---

型： List<int> (.NET) (getのみ)

解説：選択中のページ番号のリストを System.Collections.Generic.List として返します。

### 6.5.27. OpenDocument メソッド

---

**int32 OpenDocument(string filePath, string password); (.NET)**

filePath : PDF ファイルのパスを指定します。

password : パスワード パスワード無しの場合は空文字列を指定してください。

戻り値 : 成功した場合 0 が返ります。失敗した場合はエラーコードが返ります。

解説 : PDF ファイルをオープンします。 閲覧パスワードがかかっている PDF の場合、デフォルトではパスワード入力ダイアログを表示します。それ以外のエラーが発生した場合はエラーダイアログを表示します。

エラー時にダイアログを一切表示したくない場合は CtrlStyle プロパティで 0x0004 を指定してください。また、0x0008 を指定するとパスワードがかかった PDF でもパスワード入力ダイアログが表示されません。この場合パスワード処理は自分でおこなう必要があります。エラーコードが 843906 の場合は、PDF にパスワードが設定されていて、与えられた passWord が一致しないエラーです。パスワード入力を行い、再度 OpenDocument() を呼び出してください。

### 6.5.28. SaveDocument メソッド

---

**int32 SaveDocument(string filePath); (.NET)**

filePath : PDF ファイルのパスを指定します。

戻り値 : 成功した場合 0 が返ります。失敗した場合はエラーコードが返ります。

解説 : PDF ファイルを保存します。

エラー時にダイアログを一切表示したくない場合は CtrlStyle プロパティで 0x0004 を指定してください。

### 6.5.29. CloseDocument メソッド

---

**void CloseDocument(); (.NET)**

解説 : PDF ファイルをクローズします。

### 6.5.30. GetPDFDocumentHandle メソッド

---

**IntPtr GetPDFDocumentHandle();** (.NET)

戻り値： PDFDocument のハンドル

解説：コントロールに関連付けられている PDFDocument のハンドルを取得します。ハンドルは OpenDocument でファイルを開いた後に取得してください。

#### 6.5.31. AttachPDFDocument メソッド

---

**void AttachPDFDocument(IntPtr hPDFDocument);** (.NET)

hPDFDocument : PDFDocument のハンドル

解説：指定した PDFDocument のハンドルをコントロールに関連付けます。一つの PDFDocument を、複数のコントロールと使い回したいときに使用します。同じ PDFDocument を持つコントロールはアンドゥー、回転、ファイルクローズ時などを同期します。

#### 6.5.32. UnlockOwnerPasswordSecurity メソッド

---

**bool UnlockOwnerPasswordSecurity (string password);** (.NET)

password : オーナーパスワードを指定します。

戻り値：セキュリティ解除に成功した場合、true が返ります。

解説：オーナーパスワードを指定してセキュリティを解除します。

PDF にオーナーパスワードがかかっている場合、オープン時にオーナーパスワードを入力して PDF を開くか、この関数でセキュリティを解除しない限り、ページの回転、オブジェクトの追加、削除、変更といった編集機能は一切動作しません。

#### 6.5.33. GotoPage メソッド

---

**bool GotoPage (int32 pageNo);** (.NET)

pageNo : 表示するページ

戻り値：True 成功（ページが切り替わった場合）

解説：指定したページを表示します。

#### 6.5.34. GetPageRotation メソッド

---

**int32 GetPageRotation (int32 pageNo);** ([.NET](#))

pageNo : ページ番号

戻り値 : 回転角度

解説 : 指定したページの回転角度を取得します。

#### 6.5.35. RotatePages メソッド

---

**void RotatePages(int32 rotation);** ([.NET](#))

rotation : 回転角度。90/180/270 のいずれかを指定

解説 : 選択中のページを回転します。回転は 90 度単位です。

#### 6.5.36. InsertNewPages メソッド

---

**void InsertNewPages(int32 pageCount, int32 rotation, int32 pageWidth, int32 pageHeight,  
int32 destIndex);** ([.NET](#))

pageCount :挿入するページ数

rotation : 回転角度 90/180/270 のいずれかを指定

pageWidth/pageHeight : ページサイズ(twip)

destIndex : 挿入先。0 からはじまるインデックス。

解説 : 白紙ページを挿入します。

#### 6.5.37. MovePages メソッド

---

**void MovePages(int32 destIndex);** ([.NET](#))

destIndex : 移動先。0 からはじまるインデックス

解説：選択中のページを指定した位置へ移動します。

#### 6.5.38. CopyPages メソッド

---

**void CopyPages(int32 destIndex);** (.NET)

destIndex : コピー先。0 からはじまるインデックス

解説：選択中のページを指定した位置へコピーします。

#### 6.5.39. ResetView メソッド

---

**void ResetView();** (.NET)

解説：ビューをクリアして作り直します。

#### 6.5.40. Undo メソッド

---

**void Undo();** (.NET)

解説：最後の編集操作を取り消します。最高 50 回まで可能です。CanUndo()でアンドゥー可能かどうか調べることができます。

#### 6.5.41. Redo メソッド

---

**void Redo();** (.NET)

解説：Undo を取り消します。最高 50 回まで可能です。CanRedo()でリドゥー可能かどうか調べることができます。

#### 6.5.42. Cut メソッド

---

**void Cut();** (.NET)

解説：選択されたページを切り取ってクリップボードに保存します。CanCut()で切り取り可能かどうか調べることができます。

#### 6.5.43. Copy メソッド

---

**void Copy();** [\(.NET\)](#)

解説：選択されたページをコピーしてクリップボードに保存します。CanCopy()でコピー可能かどうか調べることができます

#### 6.5.44. Paste メソッド

---

**void Paste();** [\(.NET\)](#)

解説：クリップボードの内容（Cut や Copy したページ）を貼り付けます。CanPaste()で貼り付け可能かどうか調べることができます。

#### 6.5.45. Delete メソッド

---

**void Delete();** [\(.NET\)](#)

解説：選択されたページを削除します。CanDelete()で削除可能かどうか調べることができます。

#### 6.5.46. SelectPage メソッド

---

**void SelectPage(int32 pageNo, bool bSelect);** [\(.NET\)](#)

pageNo : ページ番号

bSelect : 選択する場合は True、選択解除する場合は False を指定します。

解説：指定したページを選択、または選択解除します。ページ遷移はおこないません。

#### 6.5.47. SelectAll メソッド

---

**void SelectAll(bool bSelect); (.NET)**

bSelect : 選択する場合は True、選択解除する場合は False を指定します。

解説：全ページを選択、または選択解除します。

#### 6.5.48. DocumentOpened イベント

---

**void DocumentOpened(IntPtr hPDFDocument); (.NET)**

hPDFDocument : PDFDocument のハンドル

解説：ドキュメントがオープンされたときに発行されます。

#### 6.5.49. PageTurned イベント

---

**void PageTurned(int32 pageNo); (.NET)**

pageNo : ページ番号

解説：現在ページが変更になったときに発行されます。

#### 6.5.50. DragDropFiles イベント

---

**int32 DragDropFiles(BSTR[] files); (.NET)**

files : ドロップされたファイル名

戻り値(.NET) : デフォルトの動作 (files[0]を開く) を実行する場合は 0 以外を返します。 0 を返すとデフォルト動作をおこないません。

解説：コントロールにファイルがドラッグ & ドロップされたときに発行されます。

#### 6.5.51. ContextMenuC イベント

---

**int32 ContextMenuC(System.Drawing.Point point); (.NET)**

戻り値 (.NET) : コントロールのデフォルト処理を実行する場合は 1 を返します。 0 を返すとデフォルト処理をおこないません。

point (.NET) : カーソル位置のクライアント座標が入ります。

解説：マウス右クリックされるかコンテキストメニューボタンが押されたときに発行されます。

## 7. PDFOutlineCtrl

---

### 7.1. 概要

しおりをツリー形式で表示/操作するコントロールで、.NET 版のみの提供となります。

PDFOutlineCtrl はフォームに貼り付けるだけで簡単に利用することができ、ツリーの項目をクリック、またはカーソルキーの操作でしおりに割り当てられたアクションを実行することができます。

### 7.2. アプリケーションが実装すべき項目

コントロールには多くの機能がデフォルトで実装されており、ファイルを開くだけで簡単に利用できるようになっています。

なお、PDFPageViewCtrl が複数ある場合でもしおりの操作対象となるのは 1 つの PDFPageViewCtrl のみとなります。

詳しくは付属のサンプルプログラムを参考にしてください。

機能概要	説明
コントロール間の同期	デフォルトではなにもしません。 同期をとるには DocumentOpened イベントで AttachPDFDocument してください。同期をとるとページのオープンをおこなった場合に、同期中の全コントロールに反映されます。 しおりのアクションの実行先は既定で最初に見つかった PDFPageViewCtrl となります。これを変更する場合は、AttachPDFView() メソッドを使用します。

### 7.3. しおりアクションの実行

初期状態ではしおりアクションの実行対象となるのは最初に見つかった PDFPageViewCtrl となります。

AttachPDFView() メソッドを使用することで初期状態に変わり、任意の PDFPageViewCtrl を対象とすることができます。

## 7.4. API 一覧

### 7.4.1. プロパティ

プロパティ名	.NET/.NET Framework	概要
IsOpen	○	PDF がオープンされているか調べます。
HasFocus	○	コントロールがフォーカスを持っているかどうか調べます。
DocumentPath	○	オープンしている PDF の Path を取得します。
SDKVersion	○	SDK バージョン番号を取得します。

### 7.4.2. メソッド

メソッド名	.NET/.NET Framework	概要
OpenDocument	○	PDF をオープンします。
CloseDocument	○	PDF をクローズします。
GetPDFDocumentHandle	○	PDFDocument のハンドルを取得します。
AttachPDFDocument	○	PDFDocument のハンドルをコントロールに関連付けます。
AttachPDFView	○	実行対象の実行対象となる PDFPageView を指定します。

### 7.4.3. イベント

イベント名	.NET/.NET Framework	概要
DocumentOpened	○	ドキュメントがオープンされたときに呼ばれます。

## 7.5. API リファレンス

### 7.5.1. IsOpen プロパティ

---

型： bool [\(.NET\)](#) (getのみ)

解説： PDF がオープンされているかどうか調べます。オープンされていれば True になります。

### 7.5.2. HasFocus プロパティ

---

型： bool [\(.NET\)](#) (getのみ)

解説： コントロールがフォーカスを持っているかどうか調べます。フォーカスをもっていれば True になります。.NET の Control.Focused プロパティは正しく機能しませんのでこのプロパティを使用してください。

### 7.5.3. DocumentPath プロパティ

---

型： string [\(.NET\)](#) (getのみ)

解説：オープン中 PDF ファイルの Path を取得します。

### 7.5.4. SDKVersion プロパティ

---

型： string [\(.NET\)](#) (getのみ)

解説：PDFViewerSDK のバージョン番号です。

### 7.5.5. OpenDocument メソッド

---

**int32 OpenDocument(string filePath, string password); [\(.NET\)](#)**

filePath : PDF ファイルのパスを指定します。

password : パスワード パスワード無しの場合は空文字列を指定してください。

戻り値：成功した場合 0 が返ります。失敗した場合はエラーコードが返ります。

解説： PDF ファイルをオープンします。閲覧パスワードがかかっている PDF の場合、デフォルトではパスワード入力ダイアログを表示します。それ以外のエラーが発生した場合はエラーダイアログを表示します。エラーコー

ドが 843906 の場合は、PDF にパスワードが設定されていて、与えられた passWord が一致しないエラーです。パスワード入力を行い、再度 OpenDocument()を呼び出してください。

#### 7.5.6. CloseDocument メソッド

---

**void CloseDocument();** [\(.NET\)](#)

解説：PDF ファイルをクローズします。

### 7.5.7. GetPDFDocumentHandle メソッド

---

**IntPtr GetPDFDocumentHandle();** [\(.NET\)](#)

戻り値： PDFDocument のハンドル

解説：コントロールに関連付けられている PDFDocument のハンドルを取得します。ハンドルは OpenDocument でファイルを開いた後に取得してください。

### 7.5.8. AttachPDFDocument メソッド

---

**void AttachPDFDocument(IntPtr hPDFDocument);** [\(.NET\)](#)

hPDFDocument : PDFDocument のハンドル

解説：指定した PDFDocument のハンドルをコントロールに関連付けます。一つの PDFDocument を、複数のコントロールと使い回したいときに使用します。同じ PDFDocument を持つコントロールはファイルクローズ時などを同期します。

### 7.5.9. AttachPDFView メソッド

---

**bool AttachPDFView(PDFPageViewCtrl PDFPageViewCtrl);** [\(.NET\)](#)

PDFPageViewCtrl : PDFPageViewCtrl の参照

戻り値：指定に成功した場合には true が返ります。

解説： しおりアクションの実行対象となる PDFPageViewCtrl を指定します。対象には PDFOutlineCtrl と同じ PDFDocument が関連付けされている必要があり、PDFDocument がまだ指定されていないか、異なる PDFDocument と関連付けされた PDFPageViewCtrl が指定されている場合、呼び出しは失敗します。  
なお、初期状態では最初に見つかった PDFPageViewCtrl がしおりアクションの実行対象となります。

### 7.5.10. DocumentOpened イベント

---

**void DocumentOpened(IntPtr hPDFDocument);** [\(.NET\)](#)

hPDFDocument : PDFDocument のハンドル

解説：ドキュメントがオープンされたときに発行されます。



## 8. PDFLayerCtrl

---

### 8.1. 概要

PDF のレイヤーを表示/操作するコントロールです。.NET 版のみの提供となります。

PDFLayerCtrl はフォームに貼り付けるだけで簡単に利用することができます。

レイヤーコントロールには以下の機能があります。

- ・レイヤー名の確認
- ・レイヤーの階層の確認
- ・チェックボックスでレイヤーの表示・非表示
- ・右クリックメニューから以下の操作

　　レイヤーの表示を初期状態にリセット

　　すべてを展開

　　すべてを閉じる

### 8.2. アプリケーションが実装すべき項目

コントロールには多くの機能がデフォルトで実装されており、ファイルを開くだけで簡単に利用できるようになっています。詳しくは付属のサンプルプログラムを参考にしてください。

機能概要	説明
コントロール間の同期	デフォルトではなにもしません。 同期をとるには DocumentOpened イベントで AttachPDFDocument してください。同期をとるとファイルをオープンした場合に、同期中の全コントロールに反映されます。

### 8.3. 制限事項

以下の制限事項があります。

- ・同一の PDFDocument に対して複数のレイヤーコントロールを割り当てての使用には対応しておりません。
- ・レイヤーの編集には対応していません。
- ・コントロールには全ページのレイヤーが表示されます。ページごとのレイヤー表示には対応していません。
- ・コントロールのチェックマークやラジオボタンは Windows のテーマや色の設定に同期しません。
- ・PDF 仕様における Optional Content の一部の機能には対応しておりません。

扱う Intent は「View」のみです。「Desig」は扱いません。

Image / Alternate Image / Annotation の OC キー

Usage 辞書には一部をのぞき対応していません。印刷時は表示と同じ状態になります。

複数の OCCConfig 辞書の切り替え。既定の OCCConfig（「D」）を参照します。

## 8.4. API 一覧

### 8.4.1. プロパティ

プロパティ名	.NET/.NET Framework	概要
IsOpen	○	PDF がオープンされているか調べます。
HasFocus	○	コントロールがフォーカスを持っているかどうか調べます。
DocumentPath	○	オープンしている PDF の Path を取得します。
SDKVersion	○	SDK バージョン番号を取得します。

### 8.4.2. メソッド

メソッド名	.NET/.NET Framework	概要
OpenDocument	○	PDF をオープンします。
CloseDocument	○	PDF をクローズします。
GetPDFDocumentHandle	○	PDFDocument のハンドルを取得します。
AttachPDFDocument	○	PDFDocument のハンドルをコントロールに関連付けます。
ResetView	○	ビューをクリアして作り直します。

### 8.4.3. イベント

イベント名	.NET/.NET Framework	概要
DocumentOpened	○	ドキュメントがオープンされたときに呼ばれます。

## 8.5. API リファレンス

### 8.5.1. IsOpen プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： PDF がオープンされているかどうか調べます。オープンされていれば True になります。

### 8.5.2. HasFocus プロパティ

---

型： bool [\(.NET\)](#) [\(getのみ\)](#)

解説： コントロールがフォーカスを持っているかどうか調べます。フォーカスをもっていれば True になります。.NET の Control.Focused プロパティは正しく機能しませんのでこのプロパティを使用してください。

### 8.5.3. DocumentPath プロパティ

---

型： string [\(.NET\)](#) [\(getのみ\)](#)

解説：オープン中 PDF ファイルの Path を取得します。

### 8.5.4. SDKVersion プロパティ

---

型： string [\(.NET\)](#) [\(getのみ\)](#)

解説：PDFViewerSDK のバージョン番号です。

### 8.5.5. OpenDocument メソッド

---

**int32 OpenDocument(string filePath, string password); [\(.NET\)](#)**

filePath : PDF ファイルのパスを指定します。

password : パスワード パスワード無しの場合は空文字列を指定してください。

戻り値：成功した場合 0 が返ります。失敗した場合はエラーコードが返ります。

解説： PDF ファイルをオープンします。 閲覧パスワードがかかっている PDF の場合、デフォルトではパスワード入力ダイアログを表示します。それ以外のエラーが発生した場合はエラーダイアログを表示します。 エラーコー

ドが 843906 の場合は、PDF にパスワードが設定されていて、与えられた passWord が一致しないエラーです。パスワード入力を行い、再度 OpenDocument()を呼び出してください。

#### 8.5.6. CloseDocument メソッド

---

**void CloseDocument();** [\(.NET\)](#)

解説：PDF ファイルをクローズします。

### 8.5.7. GetPDFDocumentHandle メソッド

---

**IntPtr GetPDFDocumentHandle();** [\(.NET\)](#)

戻り値： PDFDocument のハンドル

解説：コントロールに関連付けられている PDFDocument のハンドルを取得します。ハンドルは OpenDocument でファイルを開いた後に取得してください。

### 8.5.8. AttachPDFDocument メソッド

---

**void AttachPDFDocument(IntPtr hPDFDocument);** [\(.NET\)](#)

hPDFDocument : PDFDocument のハンドル

解説：指定した PDFDocument のハンドルをコントロールに関連付けます。一つの PDFDocument を、複数のコントロールと使い回したいときに使用します。同じ PDFDocument を持つコントロールはファイルクローズ時などを同期します。

### 8.5.9. ResetView メソッド

---

**void ResetView();** [\(.NET\)](#)

解説：ビューをクリアして作り直します。

### 8.5.10. DocumentOpened イベント

---

**void DocumentOpened(IntPtr hPDFDocument);** [\(.NET\)](#)

hPDFDocument : PDFDocument のハンドル

解説：ドキュメントがオープンされたときに発行されます。

## 9. 改訂履歴

---

日付	改定内容
2024 年 8 月	初版
2025 年 3 月	第 2 版 V6.0 改訂 1 版
2025 年 9 月	第 3 版 V6.0 改訂 2 版

Antenna House PDF Viewer SDK V6

PDF Viewer Ctrl インタフェース仕様

© 2024- Antenna House, Inc. All rights reserved.