Software License Agreement

PDF Report Generator

for Win32

Version 5

4380 Caldwell Palm Circle

Round Rock, TX 78665

512-733-2525

Software License Agreement

The Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software is licensed, not sold. This LICENSE AGREEMENT grants you the following rights:

A. This product is licensed per developer basis only. Each developer working with this package needs to purchase a separate license.

B. The purchaser has the right to modify and link the DLL functions into their application. Such an application is free of distribution royalties with these conditions: the target application is not a stand-alone PDF Report Generator; the target application uses this product for one operating system platform only; and the source code (or part) of the editor is not distributed in any form.

C. The DESKTOP LICENSE allows for the desktop application development. Your desktop application using this product can be distributed royalty-free. Each desktop license allows one developer to use this product on up to two development computers. A developer must purchase additional licenses to use the product on more than two development computers.

D. The SERVER LICENSE allows for the server application development. The server licenses must be purchased separately when using this product in a server application. Additionally, the product is licensed per developer basis. Only an UNLIMITED SERVER LICENSE allows for royalty-free distribution of your server applications using this product.

E. ENTERPRISE LICENSE: The large corporations with revenue more than $500 million and large government entities must purchase an Enterprise License. An Enterprise license is also applicable if any target customer of your product using the Software have revenue more than $500 million. Please contact us at info@subsystems.com for a quote for an Enterprise License.

F. Your license rights under this LICENSE AGREEMENT are non-exclusive. All rights not expressly granted herein are reserved by Licensor.

G. You may not sell, transfer or convey the software license to any third party without Licensor's prior express written consent.

This software is designed keeping the safety and the reliability concerns as the main considerations. Every effort has been made to make the product reliable and error free. However, Sub Systems, Inc. makes no warranties against any damage, direct or indirect, resulting from the use of the software or the manual and can not be held responsible for the same. The product is provided 'as is' without warranty of any kind, either expressed or implied, including but not limited to the implied warranties of suitability for a particular purpose. The buyer assumes the entire risk of any damage caused by this software. In no event shall Sub Systems, Inc. be liable for damage of any kind, loss of data, loss of profits, interruption of business or other financial losses arising directly or indirectly from the use of this product. Any liability of Sub Systems will be exclusively limited to refund of purchase price.

Sub Systems, Inc. offers a 30 day money back guarantee for the product. Must call for an RMA number before returning the product.

Desclaimer

This software is designed keeping the safety and the reliability concerns as the main considerations. Every effort has been made to make the product reliable and error free. However, Sub Systems, Inc. makes no warranties against any damage, direct or indirect, resulting from the use of the software or the manual and can not be held responsible for the same.

MSDOS, Windows 95/98/NT/2000/XP, Visual C++, MFC, and Visual Basic are the trademarks of Microsoft Corp. (for ease of reading Windows refer to MS Windows)

Delphi is the trademark of Borland International.

The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated.

Getting Started

This chapter describes the contents of the software diskettes and provides a step by step process of incorporating PDF Report Generator into your application.

In This Chapter

License Key

Your License Key and License number are e-mailed to you after your order is processed. You would set the license information using the PdgSetLicenseInfo static function. This should be preferably done before creating the converter session to avoid pop-up nag screens.

int PdgSetLicnseInfo(LPBYTE LicenseKey, LPBYTE LicenseNumber, LPBYTE CompanyName);

LicenseKey:	Your license key is available in the product delivery email sent to you upon the purchase of the product. It consists of a string in the form of "xxxxx-yyyyy-zzzzz".
LicenseNumber:	Your license number is also available in the product delivery email. The license number string starts with a "srab" or "smo" prefix.
CompanyName:	Your company name as specified in your order.

Return Value: This method returns 0 when successful. A non-zero return value indicates an error condition. Here are the possible return values:

0	License application successful.
1	Invalid License Key.
2	Invalid License Number.
3	Ran out of available licenses. Please consider purchasing additional licenses.

Example:

result=PdgSetLicenseInfo("xxxxx-yyyyy-zzzzz","srabnnnnn-n","Your Company Name")

Replace the 'xxxxx-yyyyy-zzzzz' by your license key, replace "srabnnnnn-n" with your license number, and "Your Company Name" with your company name as specified in your order.

Note: *PdgSetLicenseInfo method should be called only once at the beginning of your application. Calling this method for each conversion would degrade the conversion performance.*

Also, you can use the PdgGetLicenseStatus function at anytime to retrieve the license status.

Incorporating the DLL into Your Application

A C/C++ application should include the pdg.h and rcg.h files into the application module that needs to access the report generator. It also should include the PDG32.LIB and RCG32.LIB as the linker library. Please refer to the demo application for an example.

A Visual Basic application needs to include the PDG.BAS and RCG.BAS modules in the project. Please refer to the DMO_VB project for an example.

A Delphi application needs to include the PDG.PAS, PDG_PROP.PAS, RCG.PAS and RCG_PROT.PAS units. Please refer to the DMO_DLP project for an example

.

Creating a sample document

The following sample document would contain 2 lines. The first line is created with the bold style. The second line is centered and uses 'Arial' font. The lines are terminated with CR/LF (Ascii 13 and Ascii 10) pair to end the current paragraph.

C/C++ Example:

```
long RepId,DocId;


RepId=PdgNewSession(); // start a reporting session


DocId=PdgNewDocObject(RepId); // begin new document, and
                              //  return a  document handle


RcgInitSect(DocId); // begin first section
RcgBeginSectText(DocId); // begin the text for the section
RcgTextFont(DocId, "Times New Roman", 12, RCG_BOLD, TRUE);
RcgInsertText(DocId, "This is the first line\r\n");
RcgTextFont(DocId, "Arial", 12, RCG_BOLD, FALSE);
RcgParaFlags(DocId,RCG_CENTER);
RcgInsertText(DocId, "This is the second paragraph\r\n");
                                //  the file



PdgGenerateReportFile(RepId,DocId,"test.pdf"); //generate PDF


PdgEndSession(RepId); // free the current report handle
```

Visual Basic Example:

```
Dim RepId As Long
Dim DocId As Long
Dim result As Integer
```

```
  RepId = PdgNewSession() ' start new report session

  DocId = PdgNewDocObject(RepId) ' begin new document, and
                                 ' return a document 'handle

  result = RcgInitSect(DocId) ' begin first section
  result = RcgBeginSectText(DocId)  ' begin the text for
                                    ' the section
  result = RcgTextFont(DocId, "Times New Roman", 12, RCG_BOLD,
                       True)
  result = RcgInsertText(DocId, "This is the first line" +
         Chr$(13) + Chr$(10))   ' Ascii 13/10 pair terminates
                                'the paragraph
  result = RcgTextFont(DocId, "Arial", 12, RCG_BOLD, False)
  result = RcgParaFlags(DocId, RCG_CENTER)
  result = RcgInsertText(DocId, "This is the second paragraph"
                                + Chr$(13) + Chr$(10))


  result = PdgGenerateReportFile(RepId, DocId, "test.pdf")
                                ' generate PDF

  result = PdgEndSession(RepId) ' free the current report
handle
```

Delphi Example:

```
var
  RepId: LongInt;
  DocId: LongInt;
begin
```

```
  RepId:=PdgNewSession;  { start new reporting session }
  DocId:=PdgNewDocObject(RepId); { begin new document, and
                                   return a document handle}
  RcgInitSect(DocId); { begin first section}
  RcgBeginSectText(DocId); { begin the text for the section}
  RcgTextFont(DocId, 'Times New Roman', 12, RCG_BOLD, TRUE);
  RcgInsertText(DocId, 'This is the first line' + chr(13) +
    chr(10));      { Ascii 13/10 pair terminates the paragraph}
  RcgTextFont(DocId, 'Arial', 12, RCG_BOLD, FALSE);
  RcgParaFlags(DocId,RCG_CENTER);
  RcgInsertText(DocId, 'This is the second paragraph' +
                chr(13) + chr(10));

  PdgGenerateReportFile(RepId, DocId, 'test.pdf'); {generate
                                                    PDF}

  PdgEndSession(RepId);  {free the current report handle}

end
```

Application Interface functions

These API functions allow you to create a new reporting session, create the report contents and generate the PDF file for the report.

In This Chapter
PDG DLL Functions
RCG DLL functions

PDG DLL Functions

The PDG dll provides reporting session management functions. These functions allow you to create a new reporting session, create a new document object for the reporting session, generate the PDF file, and end the reporting session. The document object id returned by the PdgNewDocObject function is used with RCG dll functions to define the contents of the report.

In This Chapter

 

PdgEndSession

End a conversion session.

BOOL PdgEndSession(RepId)

DWORD RepId; Report session id.

Description: This function is called at the end of the generation process to free up the session related resources.

Return Value: The function returns TRUE when successful.

PdgGenerateReportBuffer

Generate the PDF data in a buffer using the RCG document object.

HGLOBAL PdgGenerateReportBuffer(RepId, DocId, OutStringLen)

DWORD RepId;	// Report session id.
DWORD DocId	// The document object which contains the coded report.
LPINT OutStringLen;	// The variable to receive the length of the converted document.

Return value: This function returns a global memory handle containing the generated PDF document. You can either use the PdgHandleToStr or GlobalLock functions to access the data string contained in this global memory handle. GlobalLock is a Windows SDK function.

A null return values indicates an error.

Examples:

```
Dim hMem as long
Dim OutSize as long

hMem = PdgGenerateReportBuffer(RepId, DocId, OutSize)
If (hMem <> 0) Then
   PdfString = Space$(OutSize + 1) ' allocate space for the
                                     output string
   PdgHandleToStr(PdfString, OutSize, hMem) ' copy pdf from
              hMem global handle to the PdfString variable.
End If
```

PdgGenerateReportFile

Generate the PDF file using the RCG document object.

BOOL PdgGenerateReportFile(RepId, DocId, OutFile)

DWORD RepId;	// Report id
DWORD DocId;	// The document object which contains the coded report.
LPBYTE OutFile;	// Output PDF file containing the report.

Return value: This method returns True when successful.

Examples:

```
PdgGenerateReportFile(RepId,DocId, "test.pdf")
```

 

PdgGetLastMessage

Get the last message.

int PdgGetLastMessage(RepId, PdgMessage, DebugMessage);

DWORD RepId;	Report session id.
LPBYTE PdgMessage;	Returns the default user message text in English
LPBYTE DebugMsg;	Returns any debug message associated with the last message. The debug message need not be displayed to the user.

Return Value: This function returns the last message generated by the editor. This value is valid only if saving of the messages is enabled by setting the PGFLAG_RETURN_MSG_ID flag. This flag is set using the PdgSetFlags function.

PdgGetLicnseStatus

Get the license status.

int PdgGetLicnseStatus()

Return Value:

0 License application successful.

1 Invalid License Key.

2 Invalid License Number.

3 Ran out of available licenses. Please consider purchasing additional licenses.

4 The evaluation period has expired.

You can use the PdgGetLicenseStatus function at anytime to retrieve the license status.

PdgHandleToStr

Convert a global memory handle to a Visual Basic string.

BOOL PdgHandleToStr(string, length, hMem)

LPBYTE string;	pointer to a visual basic string
long length	length of the string
HGLOBAL hMem;	Global memory handle

Description: This function can be used to copy the contents of a global memory handle to a given visual basic string. The calling routine must expand the string to appropriate length before calling this function.

Example:

```
string=space(length)

PdgHandleToStr(string,length,hMem)
```

The input global memory handle is freed up after copying its contents to the string.

Return Value: This function returns TRUE if successful.

PdgNewDocObject

Create a new document object for the reporting session.

DWORD PdgNewDocObject(RepId)

Description: This function creates the document object for the current reporting session. A document object id is needed to call RCG dll functions to create the content of the report.

Return Value: The function returns a non-zero session-id when successful. A zero value indicates a fail return.

PdgNewSession

Create a new reporting session.

DWORD PdgNewSession()

Description: This function needs to be called before calling any other PDG or RCG DLL functions. This function creates a new reporting session.

Normally, you would next call the PdgNewDocObject method to create a new document object.

The PdgEndSession must be called at the end to free up the session resources. All other conversion functions are called between the calls to the PdgNewSession and PdgEndSession functions.

Return Value: The function returns a non-zero session-id when successful. A zero value indicates a fail return.

PdgResetLastMessage

Reset the last editor message.

BOOL PdgResetLastMessage(RepId)

DWORD RepId; Report session id.

Description: This function can be called before calling any other function to reset the last error message.

Return Value: The function returns TRUE when successful.

PdgSetFlags

Set certain flags or retrieve the values of the flags.

DWORD PdgSetFlags(RepId, set, flags)

DWORD RepId;	Report session id.
BOOL set;	TRUE to set the given flags, FALSE to reset the given flags
DWORD flags;	Flags (bits) to set or reset. Currently, the following flag values are available:

PGFLAG_RETURN_MSG_ID	Do not display the error messages. Save the error code to be later retrieved using the PdgGetLastMessage function.

Return value: This function returns the new value of all the flags. Call this function with the 'flags' parameter set to zero to retrieve flag values without modifying it.

PdgSetBoolProp

Set a boolean property for the pdf document.

BOOL PdgSetBoolProp(RepId, prop, val)

DWORD RepId;	Report session id.
int prop;	One of the following property type to set:
PGPROP_COMPRESS_TEXT	Compress text inside the pdf document. Default = false.
PGPROP_EMBED_FONTS	Embed font data in the pdf document. Default = false.
PGPROP_BOOKMARK	Include bookmarks. Default = true.
PGPROP_HYPERLINK	Include hyperlinks. Default = true.
PGPROP_EXACT_TEXT_PLACEMENT	Use exact text placement. This flag causes the editor to emit character width for every character. Default = false.
BOOL val;	Specify TRUE or FALSE value for the specified flag.

Return value: This function returns TRUE when successful.

PdgSetNumProp

Set a numeric property for the pdf document.

BOOL PdgSetNumProp(RepId, prop, val)

DWORD RepId;	Report session id.
int prop;	One of the following property type to set:
PGPROP_PERM_FLAGS	Permission flags. The permission flags is effective only when UserPassword or OwnerPassword is also specified using the PdgSetTextProp method. One or more of the following permission flags can be specified using the 'val' parameter: PERM_PRINT — Allow printing of the document PERM_MOD — Allow modification PERM_COPY — Allow copying of the document to clipboard. You can specify more than one permission flags using the 'or' operator.
int val;	The numeric value of the selected property.

Return value: This function returns TRUE when successful.

PdgSetTextProp

Set a text property for the pdf document.

BOOL PdgSetTextProp(RepId, prop, val)

DWORD RepId;	Report session id.
int prop;	One of the following property type to set:

PGPROP_AUTHOR	Author of the document
PGPROP_TITLE	Document title
PGPROP_SUBJECT	Subject
PGPROP_KEYWORDS	Key words
PGPROP_CRE_DATE	Creation date
PGPROP_MOD_DATE	Modification date
PGPROP_USER_PASSWORD	User password
PGPROP_OWNER_PASSWORD	Owner password

LPBYTE val;	The text value of the selected property.

Return value: This function returns TRUE when successful.

RCG DLL functions

These API functions allow create the content of the report. Your application must create a document id by using the PdgNewDocObject function exported by PDG dll. All RCG dll functions require a document id as the first parameter.

In This Chapter

RcgResetTextProp
RcgResetParaProp
RcgSectInfo
RcgSetFlags
RcgTextColor
RcgTextFont

RcgBeginCellText

Begin the text for a cell.

BOOL RcgBeginCellText(DocId)

DWORD DocId; The id of the current document being generated.

Description: Any text or graphic within a cell must be placed between the RcgBeginCellText and RcgEndCellText calls. The first call for RcgBeginCellText for a row would occur after calling the RcgNextCellInfo function for all cells in the row. The RcgBeginCellText call for the next cell would occur after the RcgEndCellText call for the previous cell in the row. Please refer to the demo program for an example of calling this function.

Return Value: This function returns TRUE if successful.

See Also:
RcgEndCellText

RcgBeginFrame

Begin a positionable frame.

BOOL RcgBeginFrame(DocId,PageRelative,x, y,width,height)

DWORD DocId;	The id of the current document being generated.
BOOL PageRelative;	Set to TRUE to create a frame relative to the top of the page. Set to FALSE to create a frame relative to the current paragraph.
int x;	The x position in twips.
int y;	The y position in twips
int width;	The width of the frame in twips.
int height	The height of the frame in twips.

Description: This function begins a positionable frame. Any call to the text or graphic insertion functions after this function call would place the text or graphic inside the frame. The text outside the frame flows around the frame. When done, call the RcgEndFrame function to end the frame.

Return Value: This function returns TRUE if successful.

See Also:
RcgEndFrame

 

RcgBeginGroup

Begin an RTF group.

BOOL RcgBeginGroup(DocId,name,IsDest)

DWORD DocId;	The id of the current document being generated.
LPBYTE name;	The name of the group
BOOL IsDest;	Set to TRUE to create a destination group. An entire destination group is ignored by an RTF reader if it does not support the group.

Description: This function is useful to manually add unsupported RTF features into the document. You would typically build a group manually by using the RcgInsertControl and RcgInsertText functions after calling this function. When done, the group must be closed using the RcgEndGroup function.

Return Value: This function returns TRUE if successful.

See Also:
RcgEndGroup
RcgInsertControl

RcgBeginHdrFtr

Begin a page header/footer group.

BOOL RcgBeginHdrFtr(DocId,type)

DWORD DocId; The id of the current document being generated.

int type The header/footer type:

```
RCG_HDR:   Regular header

RCG_FTR:   Regular footer

RCG_FHDR:  First page header

RCG_FFTR:  First page footer

RCG_LHDR:  Left page header

RCG_LFTR:  Left page footer

RCG_RHDR:  Right page header

RCG_RFTR:  Right page footer
```

Description: This function begins a header or a footer group. Call the text and graphic insertion functions to place text and graphic inside this group. When done, call the RcgEndHdrFtr function to end the group.

The header/footer information resides in the section initialization area. Therefore, this function is valid only after calling the RcgInitSect function and before calling RcgBeginSectText function.

Return Value: This function returns TRUE if successful.

See Also:
RcgEndHdrFtr

RcgBeginSectText

Begin the text for the current section.

BOOL RcgBeginSectText(DocId)

DWORD DocId; The id of the current document being generated.

Description: This function begins the text or the body of the section. It is typically called after the section initialization is complete. The section initialization process consists of calling the RcgInitSect function and optionally followed by RcgSectInfo, RcgBeginHdrFtr, RcgBeginStylesheet, RcgPaper, RcgMargin function. In a simplest document, RcgBeginSectText would followed immediately after the RctInitSect function.

Return Value: This function returns TRUE if successful.

See Also:
RcgInitSect

RcgBeginStyleItem

Begin a stylesheet item.

BOOL RcgBeginStyleItem(DocId,ParaStyle,StyleId,name)

DWORD DocId;	The id of the current document being generated.
BOOL ParaStyle;	TRUE to create a paragraph style, or FALSE to create a character style.
int StyleId;	A numeric style id. The first style should be given the id of 0, the second the id of 1, and so on.
LPBYTE name	The style name.

Description: This function begins a style item. It should be followed by the paragraph (such as RcgParaIndent and RcgParaSpace) and character formatting functions (such as RcgTextFont and RcgTextColor). A paragraph style can use both paragraph and character formatting attributes, but a character style can use only the character formatting attributes. When done, call the RcgEndStyleItem function.

The first two styles in the stylesheet should have the predefined id and name. The first style should be a paragraph style with the id of 0 and the name 'Normal'. The second style should be a character style with the id of 1 and the name 'Default Paragraph Font'. The subsequent style would use the next sequential id and a unique name.

This function is valid only inside the stylesheet group which is begun using the RcgBeginStyleSheet function.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginStyleSheet
RcgEndStyleItem

RcgBeginStyleSheet

Begin the stylesheet group.

BOOL RcgBeginStyleSheet(DocId)

DWORD DocId; The id of the current document being generated.

Description: This function begins the stylesheet group. There can be only one stylesheet group in an RTF file. If a stylesheet is desired in the RTF file, this function should be called after calling the RcgInitSect function but before calling the RcgBeginSectText function. To begin a style item, call the RcgBeginStyleItem function immediately after calling the RcgBeginStyleSheet function. When done creating all style items, call the RcgEndStyleSheet function to end the stylesheet group.

Return Value: This function returns TRUE if successful.

See Also:
RcgEndStyleSheet
RcgBeginStyleItem

RcgBeginTableRow

Begin a table row.

BOOL RcgBeginTableRow(DocId,indent,height,just,IsHdr,CellMargin)

BOOL RcgBeginTableRow2(DocId,indent,height,just,IsHdr,LeftMargin. RightMargin, TopMargin, BotMargin)

DWORD DocId;	The id of the current document being generated.
int indent;	The left indentation of the table row in twips unit. Set to 0 for default.
int height;	Use a positive value (in twips) for this parameter to specify the *minimum* height for the table row. Use a negative value (in twips) to specify the *exact* height for the table row. Set to 0 for default.
int just;	Set to 0 for default, or use one of the following values: RCG_CENTER: Center the table row RCG_RIGHT: Right justify the table row
BOOL IsHdr;	Set to TRUE to repeat this row on the next page if the table overflows the page. Set to FALSE for default.
int CellMargin;	The distance of the text in the cell to the left and right edges of the cell. Set to 0 for default. This parameter is used by the RcgBeginTableRow method only.
int LeftMargin;	The distance of the text in the cell to the left edge of the cell.
int RightMargin;	The distance of the text in the cell to the right edge of the cell.
int TopMargin;	The distance of the text in the cell to the top edge of the cell.
int BotMargin;	The distance of the text in the cell to the bottom edge of the cell.

Description: This function begins a table row. This function should be followed by the RcgNextCellInfo function for each cell in the table. Please refer to the demo program for an example of using this function.

Return Value: This function returns TRUE if successful.

See Also:
RcgEndTableRow
RcgNextCellInfo

RcgBeginTextBox

Begin a text box group.

BOOL RcgBeginTextBox(DocId,PageRelative,x,y,width,height,ZOrder,
FillColor,BorderColor,BorderWidth,Xparent)

DWORD DocId;	The id of the current document being generated.
BOOL PageRelative;	Set to TRUE to create the text box relative to the top of the page, or set to FALSE to create the text box relative to the next paragraph.
int x;	The x position of the text box in twips.
int y;	The y position of the text box in twips.
int width;	The width of the text box in twips.
int height;	The height of the text box in twips.
int ZOrder;	The Z order of the text box. Set to 0 for default.
COLORREF FillColor;	The background color of the text box. Set to hex FFFFFF for default.
COLORREF BorderColor;	The color of the border. Set to 0 for default.
int BorderWidth;	The width of the border in twips.
BOOL Xparent	Set to TRUE to create a transparent text box. Set to FALSE for default.

Description: This function begins a text box. This function can be followed by any text insertion, and character/paragraph attribute functions. When done, call the RcgEndTextBox function to end the text box.

Return Value: This function returns TRUE if successful.

See Also:
RcgEndTextBox
RcgDrawRect
RcgDrawLine
RcgBeginFrame

RcgCharStyleId

Specify a character style id for the character.

BOOL RcgCharStyleId(DocId,id)

DWORD DocId;	The id of the current document being generated.
int id;	The id of a character style item. This id must be one of style ids already created using the RcgBeginStyleItem function.

Description: When the character style id is used for character formatting, the corresponding character attributes in the style must also be specified for the text using the character formatting functions such as RcgTextFont and RcgTextColor.

Return Value: This function returns TRUE if successful.

See Also:
RcgParaStyleId
RcgBeginStyleItem

RcgDrawLine

Add an item to the selection box.

BOOL RcgDrawLine(DocId,PageRelative,x1,y1,x2,y2,ZOrder,LineColor, LineWidth)

DWORD DocId;	The id of the current document being generated.
BOOL PageRelative;	Set to TRUE to create the line object relative to the top of the page, or set to FALSE to create the line object relative to the next paragraph.
int x1;	The beginning x position of the line object in twips.
int y1;	The beginning y position of the line object in twips.
int x2;	The ending x position of the line object in twips.
int y2;	The ending y position of the line object in twips.
int ZOrder;	The Z order of the line object. Set to 0 for default.
COLORREF LineColor;	The color of the line object. Set to 0 for default.
int LineWidth;	The line width in twips.

Return Value: This function returns TRUE if successful.

See Also:
RcgDrawRect

RcgDrawRect

Add an item to the selection box.

BOOL RcgDrawRect(DocId,PageRelative,x,y,width,height,ZOrder,FillColor,
BorderColor,BorderWidth,Xparent)

DWORD DocId;	The id of the current document being generated.
BOOL PageRelative;	Set to TRUE to create the rectangle object relative to the top of the page, or set to FALSE to create the rectangle object relative to the next paragraph.
int x;	The x position of the rectangle object in twips.
int y;	The y position of the rectangle object in twips.
int width;	The width of the rectangle object in twips.
int height;	The height of the rectangle object in twips.
int ZOrder;	The Z order of the rectangle object. Set to 0 for default.
COLORREF FillColor;	The background color of the rectangle object. Set to hex FFFFFF for default.
COLORREF BorderColor;	The color of the border. Set to 0 for default.
int BorderWidth;	The width of the border in twips.
BOOL Xparent	Set to TRUE to create a transparent rectangle object. Set to FALSE for default.

Return Value: This function returns TRUE if successful.

See Also:
RcgDrawLine
RcgBeginTextBox

 

RcgEndCellText

End the text for a cell.

BOOL RcgEndCellText(DocId)

DWORD DocId; The id of the current document being generated.

Description: Any text or graphic within a cell must be placed between the RcgBeginCellText and RcgEndCellText calls. The RcgEndCellText call for the last cell in the row is followed by RcgEndTableRow function call. Please refer to the demo program for an example of calling this function.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginFrame

RcgEndFrame

End the current frame.

BOOL RcgEndFrame(DocId)

DWORD DocId; The id of the current document being generated.

Return Value: This function returns TRUE if successful.

 

RcgEndGroup

End the current RTF group.

BOOL RcgEndGroup(DocId)

DWORD DocId; The id of the current document being generated.

Description: This function is used to close an RTF group which was created using the RcgBeginGroup function.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginGroup

RcgEndHdrFtr

End the current header/footer group.

BOOL RcgEndHdrFtr(DocId)

DWORD DocId; The id of the current document being generated.

Comment: The last paragraph must he duly terminated before calling this function. To terminate an unterminated paragraph, insert a string containing ASCII 13 and ASCII 10.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginHdrFtr

RcgEndStyleItem

End the current style item.

BOOL RcgEndStyleItem(DocId)

DWORD DocId; The id of the current document being generated.

Description: To begin the next style item, call the RcgBeginStyleItem function immediately after calling this function. To end the stylesheet group, call the RcgEndStyleSheet function immediately after calling this function.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginStyleItem
RcgEndStyleSheet

 

RcgEndStyleSheet

End the stylesheet group.

BOOL RcgEndStyleSheet(DocId)

DWORD DocId; The id of the current document being generated.

Description: This function is called immediately after ending the last style item using the RcgEndStyleItem function.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginStyleSheet
RcgEndStyleItem

RcgEndTableRow

End a table row.

BOOL RcgEndTableRow(DocId)

DWORD DocId; The id of the current document being generated.

Description: This function terminates a table row. This function follows after the RcgEndCellText function for the last table row.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginTableRow

RcgEndTextBox

End a text box group.

BOOL RcgEndTextBox(DocId)

DWORD DocId; The id of the current document being generated.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginTextBox

RcgGetData

Retrieve the RTF code.

HGLOBAL RcgGetData(DocId,len)

DWORD DocId; The id of the current document being generated.

LPLONG len; The pointer to receive the byte length of the generated RTF code.

Description: This function can be used before calling the RcgEndDoc function. This function returns the RTF code in a memory handle.

A Visual Basic application can use the RcgHandleToStr function to convert the memory handle to a Basic string.

The memory handle returned by this function can also be passed to TE Edit Control using the SetTerBuffer function. TE Edit Control is a separate product (an advanced RTF Edit control) offered by Sub Systems, Inc.

Return Value: This function returns TRUE if successful

See Also:
RcgHandleToStr

RcgGetLastMessage

Get the last message.

int RcgGetLastMessage(RcgMessage, DebugMessage);

LPBYTE RcgMessage;	// Returns the default user message text in English
LPBYTE DebugMsg;	// Returns any debug message associated with the last message. The debug message need not be displayed to the user.

Return Value: This function returns the last message generated by the editor. This value is valid only if saving of the messages is enabled by setting the RFLAG_RETURN_MSG_ID flag. This flag is set using the RcgSetFlags function.

The message string constants (MSG_) are defined in the RCG.H, RCG.BAS and RCG.PAS files. The description for the message ids can be found in the RCG_MSG.H file.

See Also

RcgResetLastMessage

RcgHandleToStr

Convert a global memory handle to a Visual Basic string.

BOOL RcgHandleToStr(string, length, hMem)

LPBYET string;	pointer to a visual basic string
long length	length of the string
HGLOBAL hMem;	Global memory handle

Description: This function can be used to copy the contents of a global memory handle to a given visual basic string. The calling routine must expand the string to appropriate length before calling this function.

Example:

```
string=space(length)

HandleToStr(string,length,hMem)
```

The input global memory handle is freed up after copying its contents to the string.

Return Value: This function returns TRUE if successful.

See Also:
RcgGetData

RcgInitSect

Initialize a section.

BOOL RcgInitSect(DocId)

DWORD DocId; The id of the current document being generated.

Description: This function initializes a section. This function must be called immediately after calling the RcgNewDoc function to initialize the first section. It must also be called immediately after calling the RcgInsertBreak function to initialize the new section. Any section information functions such as RcgSectInfo, RcgBeginHdrFtr or RcgPaper must be called after calling the RcgInitSect function, but before calling the RcgBeginSectText function.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginSectText

RcgInsertBookmark

Insert a bookmark.

BOOL RcgInsertBookmark(DocId, name, start)

DWORD DocId;	The id of the current document being generated.
LPBYTE name;	The bookmark name. A space character is not allowed in the bookmark name.
BOOL start;	Set to TRUE to start a bookmark. Set to FALSE to end a previously started bookmark. Every bookmark should be ended properly by calling this function.

Return Value: This function returns TRUE if successful.

See Also:
RcgInsertText

RcgInsertBreak

Insert a text break in the document.

BOOL RcgInsertBreak(DocId,type)

DWORD DocId;	The id of the current document being generated.
int type;	Please use one of the constants to specify the text break type:
	RCG_BREAK_PAGE: Page Break
	RCG_BREAK_COL: Column break
	RCG_BREAK_SECT: Section Break.

Description: If a section break is created using this function, then RcgInitSect function must follow immediately to initialize the new section.

Return Value: This function returns TRUE if successful.

See Also:
RcgInitSect

RcgInsertControl

Insert an RTF control word.

BOOL RcgInsertControl(DocId,name,UseParam,param)

DWORD DocId;	The id of the current document being generated.
LPBYTE name;	The name of the control
BOOL UseParam;	Set to TRUE if the control uses a parameter.
long param;	The numeric value of the control parameter. This parameter is ignored if UseParam is set to FALSE.

Description: This function is used to manually insert an RTF control word.

Return Value: This function returns TRUE if successful.

See Also:
RcgBeginGroup

RcgInsertFootnote

Insert footnote text.

BOOL RcgInsertFootnote(DocId, marker, text)

DWORD DocId;	The id of the current document being generated.
LPBYTE marker;	The footnote number text. Set this parameter to NULL to specify auto footnote numbering.
LPBYTE text;	The footnote text.

Description: This function can be called within any group that allows for text insertion.

Return Value: This function returns TRUE if successful.

RcgInsertMergeField

Insert a mail-merge field.

BOOL RcgInsertMergeField(DocId,name,data)

DWORD DocId; The id of the current document being generated.

LPBYTE name; field name

LPBYTE data; initial data

Description: This function can be called within any group that allows for text insertion.

Return Value: This function returns TRUE if successful.

RcgInsertPageField

Insert page number or page count.

BOOL RcgInsertPageField(DocId,InsertPageNo)

DWORD DocId;	The id of the current document being generated.
BOOL InsertPageNo;	Set to TRUE to insert the page number field, or set to FALSE to insert page count field.

Description: This function is typically called inside a header/footer group, but it can be called within any group that allows for text insertion.

Return Value: This function returns TRUE if successful.

RcgInsertPictFile

Insert a picture.

BOOL RcgInsertPictFile(DocId,file,width,height,linked)

DWORD DocId;	The id of the current document being generated.
LPBYTE file;	The picture file name or full path of the picture.
int width;	The picture width in twips. Set to 0 to use the actual width of the picture.
int height;	The picture height in twips. Set to 0 to use the actual height of the picture.
BOOL linked	Set to TRUE to create a link to the picture file instead of embedding it inside the RTF document.

Description: Currently the DLL supports the BMP and WMF picture formats for embedding. It supports all picture formats when the picture is linked instead of embedded.

Return Value: This function returns TRUE if successful.

RcgInsertRaw

Insert raw RTF code into the document.

BOOL RcgInsertRaw(DocId,code)

DWORD DocId; The id of the current document being generated.

LPBYTE code; The rtf code text to be inserted.

Description: This function inserts the rtf code without any character translation.

Return Value: This function returns TRUE if successful.

RcgInsertText

Insert text into the document.

BOOL RcgInsertText(DocId,text)

DWORD DocId;	The id of the current document being generated.
LPBYTE text;	The text to be inserted. To end a paragraph, terminate the text string with a cr/lf (Ascii 13 and Ascii 10) pair of characters.

Description: The RcgBeginSectText function must have already been called for this function (as well any character and paragraph formatting functions) to succeed.

Return Value: This function returns TRUE if successful.

RcgInsertTocField

Insert Table-of-Content field.

BOOL RcgInsertTocField(DocId,FirstLevel,LastLevel)

DWORD DocId;	The id of the current document being generated.
int FirstLevel;	The first header level to be extracted to build table-of-contents. This parameter value must be between 1 and 9.
int LastLevel;	The last header level to be extracted to build table-of-contents. This parameter value must be between 1 and 9. The LastLevel value must be equal to or greater than the FirstLevel parameter value.

Description: This function is use to insert a table-of-content field. An RTF reader would create a table-of-content by extracting the paragraph text which uses the paragraph style names 'heading 1', 'header 2', etc.. Therefore, you would need to use the RcgBeginStyleSheet function to create such paragraph styles for the document. You would then apply these style ids to the selected text as they are appended to the document. You can use the RcgParaStyleId function to apply the paragraph style to the paragraph being added.

Optionally, you can also create the style names 'toc N' (such as 'toc 1', 'toc 2', etc.). These styles are used to create the text lines for the table-of-content. For example, an RTF reader would apply the style name 'toc 1' to the document text which uses the style name 'heading 1' to create a line for the table-of-content'. If you do not create these styles, then an RTF reader would create these styles for your document by use using the default values for font and paragraph attributes.

Return Value: This function returns TRUE if successful.

See Also

RcgBeginStyleSheet
RcgBeginStyleItem
RcgParaStyleId

RcgInsertUText

Insert Unicode text into the document.

BOOL RcgInsertUText(DocId,text)

DWORD DocId;	The id of the current document being generated.
LPWORD text;	The text to be inserted. To end a paragraph, terminate the text string with a cr/lf (Ascii 13 and Ascii 10) pair of characters.

Description: The RcgBeginSectText function must have already been called for this function (as well any character and paragraph formatting functions) to succeed.

Return Value: This function returns TRUE if successful.

RcgMargin

Set the margin for the current section.

BOOL RcgMargin(DocId,LeftMargin,RightMargin,TopMargin,BotMargin,
HdrMargin,FtrMargin)

DWORD DocId;	The id of the current document being generated.
int LeftMargin;	The left margin in twips.
int RightMargin;	The right margin in twips.
int TopMargin;	The top margin in twips.
int BotMargin;	The bottom margin in twips.
int HdrMargin;	The distance of the page header from the top of the page. Set to 0 for default.
int FtrMargin;	The distance of the page footer from the bottom of the page. Set to 0 for default.

Description: This function is called between RcgInitSect and RcgBeginSectText functions.

Return Value: This function returns TRUE if successful.

See Also:
RcgInitSect
RcgSectInfo
RcgBeginSectText
RcgPaper

RcgNextCellInfo

Specify the attributes of a table cell.

BOOL RcgNextCellInfo(DocId,width,valign,shading,BackColor,LeftWidth,
RightWidth,TopWidth,BotWidth,MergeFlags)

BOOL RcgNextCellInfo2(DocId,width,valign,shading,BackColor,LeftWidth,
RightWidth,TopWidth,BotWidth,BorderColor,MergeFlags)

DWORD DocId;	The id of the current document being generated.
int width;	The width of the cell in twips.
int valign;	Vertical alignment of the text inside the cell. Use one of the following constants or set to 0 for default.
	RCG_VALIGN_CTR: Center the text vertically.
	RCG_VALIGN_BOT: Align the text to the bottom of the cell.
int shading;	Cell shading percentage. Set to 0 for default.
COLORREF BackColor;	Background color for the cell. Set to hex FFFFFF for default.
int LeftWidth;	The width of the left edge of the cell in twips.
int RightWidth;	The width of the right edge of the cell in twips.
int TopWidth;	The width of the top edge of the cell in twips.
int BotWidth;	The width of the bottom edge of the cell in twips.
COLORREF BorderColor;	Cell border color.
UINT MergeFlags;	Set to 0 for default, or use one of the following constants:
	RCG_MRG_ROW_FIRST: The first cell to be merged vertically.
	RCG_MRG_ROW: The subsequent cell to be merged vertically.
	RCG_MRG_COL_FIRST: The first cell to be merged horizontally.
	RCG_MRG_COL: The subsequent cell to be merged horizontally.

Description: Typically this function is called after calling the RcgBeginTableRow function. This function is called for each cell in the row before calling the RcgBeginCellText function for the first cell in the row.

Return Value: This function returns TRUE if successful.

See Also:
RcgInitSect

RcgPaper

Set the paper dimension for the current section.

BOOL RcgPaper(DocId,width,height,landscape)

DWORD DocId;	The id of the current document being generated.
int width;	The paper width in twips.
int height;	The paper height in twips.
BOOL landscape;	Set to TRUE to use the landscape orientation.

Description: This function is called between RcgInitSect and RcgBeginSectText function.

Return Value: This function returns TRUE if successful.

See Also:
RcgInitSect
RcgSectInfo
RcgBeginSectText
RcgMargin

RcgParaBullet

Create paragraph bullet or paragraph numbering.

BOOL RcgParaBullet(DocId,IsBullet,type,level,BefText,AftText,flags)

DWORD DocId;	The id of the current document being generated.
BOOL IsBullet;	Set to TRUE to create the paragraph bullet, or set to FALSE to create the paragraph numbering.
int type;	Specify one of the following constants if the 'IsBullet' argument is set to TRUE::

BLT_ROUND	Round bullet
BLT_DIAMOND:	Diamond shaped bullet
BLT_SQUARE:	Square shaped bullet
BLT_HOLLOW_SQUARE:	Hollow square
BLT_4_DIAMONDS:	Four diamond shaped bullet
BLT_ARROW:	Arrow shaped bullet
BLT_CHECK:	Check mark bullet

Specify one constants if the 'IsBullet' argument is set to FALSE:

NBR_DEC:	Decimal numbering
NBR_UPPER_ALPHA:	Upper alphabetic characters.
NBR_LOWER_ALPHA:	Lower alphabetic characters.

int level;	Bullet level. Set to 0 for default.
int start;	The starting paragraph number. Set to 1 for default.
LPBYTE BefText;	The text before the paragraph number. Set to NULL for default.
LPBYTE AftText;	The text after the paragraph number. Set to NULL for default.
UINT flags;	Set to BLTFLAG_HIDDEN to hide the paragraph numbering. Set to 0 for default.

Description: This paragraph formatting set by this function is effective for the subsequent RcgInsertText function. The paragraph formatting can be reset by using the RcgResetParaProp function.

Return Value: This function returns TRUE if successful.

See Also:

RcgParaFlags
RcgParaIndent
RcgParaShadeColor
RcgParaSpace
RcgParaStyleId
RcgParaTabStop

RcgParaFlags

Set the paragraph flags.

BOOL RcgParaFlags(DocId,flags)

DWORD DocId;	The id of the current document being generated.	
DWORD flags;	Use one or more of the following constants:	
	RCG_CENTER:	Center the paragraph.
	RCG_RIGHT:	Right justify the paragraph.
	RCG_JUSTIFY:	Left/right justify the paragraph.
	RCG_DOUBLE_SPACE:	Double space the paragraph.
	RCG_PARA_KEEP:	Keep the entire paragraph in the same page.
	RCG_PARA_KEEP_NEXT:	Keep the last line of the paragraph and the first line of the next paragraph in the same page.
	RCG_PARA_WIDOW:	Keep the first 2 lines or the last 2 lines of the paragraph in the same page.
	RCG_BOX_TOP:	Create the top paragraph border.
	RCG_BOX_BOT:	Create the bottom paragraph border.
	RCG_BOX_LEFT:	Create the left paragraph border.
	RCG_BOX_RIGHT:	Create the right paragraph border.
	RCG_BOX_DOUBLE:	Create the double line paragraph border.
	RCG_BOX_THICK:	Create the thick line paragraph border.
	Please use the logical OR operator to specify more than more constants	

Description: This paragraph formatting set by this function is effective for the subsequent RcgInsertText function. The paragraph formatting can be reset by using the RcgResetParaProp function.

Return Value: This function returns TRUE if successful.

See Also:

RcgParaBullet
RcgParaIndent
RcgParaShadeColor
RcgParaSpace
RcgParaStyleId
RcgParaTabStop

RcgParaIndent

Set the paragraph indentation.

BOOL RcgParaIndent(DocId,LeftIndent,RightIndent,FirstIndent)

DWORD DocId;	The id of the current document being generated.
int LeftIndent;	The left indentation in twips. Set to 0 for default.
int RightIndent;	The right indentation in twips. Set to 0 for default.
int FirstIndent;	The additional indentation for the first line of the paragraph. Set to 0 for default.

Description: This paragraph formatting set by this function is effective for the subsequent RcgInsertText function. The paragraph formatting can be reset by using the RcgResetParaProp function.

Return Value: This function returns TRUE if successful.

See Also:

RcgParaFlags
RcgParaBullet
RcgParaShadeColor
RcgParaSpace
RcgParaStyleId
RcgParaTabStop

RcgParaShadeColor

Set the paragraph shading or paragraph background color.

BOOL RcgParaShadeColor(DocId,shade,BkColor)

DWORD DocId;	The id of the current document being generated.
int shade;	The paragraph shading in percentage. Set to 0 to set paragraph background color instead.
COLORREF BkColor;	The background color for the paragraph. This argument is ignored when the 'shade' argument is non-zero.

Description: This paragraph formatting set by this function is effective for the subsequent RcgInsertText function. The paragraph formatting can be reset by using the RcgResetParaProp function.

Return Value: This function returns TRUE if successful.

See Also:
RcgParaFlags
RcgParaIndent
RcgParaBullet
RcgParaSpace
RcgParaStyleId
RcgParaTabStop

RcgParaSpace

Set paragraph spacing.

BOOL RcgParaSpace(DocId,SpaceBef,SpaceAft,SpaceBet)

DWORD DocId;	The id of the current document being generated.
int SpaceBef;	The space before the paragraph in twips. Set to -1 to leave this value unchanged.
int SpaceAft;	The space after the paragraph in twips. Set to -1 to leave this value unchanged.
int SpaceBet;	A positive value specifies the minimum space between the paragraph lines. Set to -1 to leave this value unchanged. Any other negative value specifies exact line spacing.

Description: This paragraph formatting set by this function is effective for the subsequent RcgInsertText function. The paragraph formatting can be reset by using the RcgResetParaProp function.

Return Value: This function returns TRUE if successful.

See Also:
RcgParaFlags
RcgParaIndent
RcgParaShadeColor
RcgParaBullet
RcgParaStyleId
RcgParaTabStop

RcgParaStyleId

Specify a paragraph style id for the paragraph.

BOOL RcgCharStyleId(DocId,id)

DWORD DocId;	The id of the current document being generated.
int id;	The id of a paragraph style item. This id must be one of style ids already created using the RcgBeginStyleItem function.

Description: When the paragraph style id is used for paragraph formatting, the corresponding paragraph attributes in the style must also be specified for the text using the paragraph formatting functions such as RcgTextFont and RcgTextColor.

Return Value: This function returns TRUE if successful.

See Also:
RcgCharStyleId
RcgBeginStyleItem

RcgParaTabStop

Set a tab stop for a paragraph.

BOOL RcgParaTabStop(DocId,type,pos,flags)

DWORD DocId;	The id of the current document being generated.	
int type;	The tab stops type:	
	TAB_LEFT:	Left tab.
	TAB_RIGHT:	Right tab
	TAB_CENTER:	Center tab.
	TAB_DECIMAL:	Decimal tab.
int pos;	The tab stop position in twips.	
UINT flags	The tab leader type:	
	TAB_DOT:	Dot leader
	TAB_HYPH:	Hyphen leader
	TAB_ULINE:	Underline leader.
	Or, set to 0 for default.	

Description: When a paragraph uses multiple tab stops, the tab stops must be created in the ascending tab position order.

This paragraph formatting set by this function is effective for the subsequent RcgInsertText function. The paragraph formatting can be reset by using the RcgResetParaProp function.

Return Value: This function returns TRUE if successful.

See Also:
RcgParaFlags
RcgParaIndent
RcgParaShadeColor
RcgParaSpace
RcgParaStyleId
RcgParaBullet

RcgPastePicture

Paste a picture from clipboard for memory buffer.

BOOL RcgPastePicture(DocId,format, hData,width,height)

DWORD DocId;	The id of the current document being generated.
int format;	The format of the picture data:
	CF_DIB: Device independent bitmap
	CF_METAFILEPICT: Windows metafile
	CF_ENHMETAFILE: Enhanced metafile (Win32)
HGLOBAL hData;	The picture data in the format specified by the 'format' argument. Set the format and hData to NULL to paste the picture from clipboard.
int width;	The picture width in twips. Set to 0 to use the actual width of the picture.
int height;	The picture height in twips. Set to 0 to use the actual height of the picture.

Return Value: This function returns TRUE if successful.

RcgResetLastMessage

Reset the last editor message.

BOOL RcgResetLastMessage()

Description: This function can be called before calling any other function to reset the last error message.

Return Value: The function returns TRUE when successful.

See Also
RcgGetLastMessage
RcgSetFlags

RcgResetTextProp

Reset the character formatting attributes.

BOOL RcgResetTextProp(DocId)

DWORD DocId; The id of the current document being generated.

Description: This function is typically called before setting new character formatting attributes.

Return Value: This function returns TRUE if successful.

See Also:
RcgTextFont

RcgResetParaProp

Reset the paragraph formatting attributes.

BOOL RcgResetParaProp(DocId)

DWORD DocId; The id of the current document being generated.

Description: This function is typically called before setting new paragraph formatting attributes.

Return Value: This function returns TRUE if successful.

See Also:
RcgParaFlags
RcgParaIndent
RcgParaShadeColor
RcgParaSpace
RcgParaStyleId
RcgParaBullet

RcgSectInfo

Set the section information.

BOOL RcgSectInfo(DocId,columns,ColSpace,NewPage,FirstPageNo)

DWORD DocId;	The id of the current document being generated.
int columns;	The number of columns for the section. Set to 1 for default.
int ColSpace;	Space between the columns. Set to 0 for default.
BOOL NewPage;	Set to TRUE to begin this section on the new page.
int FirstPageNo;	The first page number for the section. Set to 0 to use continuous page numbering.

Description: This function can be used immediately after the RcgInitSect function to specify new attributes for the section.

Return Value: This function returns TRUE if successful.

See Also:
RcgInitSect
RcgMargin
RcgPaper

RcgSetFlags

Set certain flags or retrieve the values of the flags.

DWORD RcgSetFlags(DocId, set, flags)

DWORD DocId	// The DocId to be accessed. If DocId is set to 0, this function sets the common initialization flags. When a new document is created, it inherits all common flags.
BOOL set;	// TRUE to set the given flags, FALSE to reset the given flags
DWORD flags;	// Flags (bits) to set or reset. Currently, the following flag values are available:

RFLAG_RETURN_MSG_ID	Do not display the error messages. Save the error code to be later retrieved using the RcgGetLastMessage function.
RFLAG_FIRST_MSG_ONLY	Only display the first error message.

Return value: This function returns the new value of all the flags. Call this function with the 'flags' parameter set to zero to retrieve flag values without modifying it.

RcgTextColor

Specify text color.

BOOL RcgTextColor(DocId,CurColor,foreground)

DWORD DocId;	The id of the current document being generated.
COLORREF CurColor	The foreground or background text color.
BOOL foreground;	Set to TRUE to set the foreground color, or FALSE to set the background color.

Description: This text color set by this function is effective for the subsequent RcgInsertText function. The text color can be reset by using the RcgResetTextProp function.

Return Value: This function returns TRUE if successful.

See Also:
RcgTextFont

RcgTextFont

Set the text font.

BOOL RcgTextFont(DocId,typeface,PointSize,style,set)

DWORD DocId;	The id of the current document being generated.
LPBYTE typeface;	The font typeface. Set to NULL or "" to leave this attribute unchanged.
int PointSize;	The point size for the text. Set to 0 to leave this attribute unchanged. Use negative sign to specify half-points. For example, to specify a value of 8.5 points, pass -17 for this parameter.
UINT style;	Use one or more of the following style constants: `RCG_ULINE:` Underline `RCG_BOLD:` Bold `RCG_ITALIC:` Italic `RCG_HIDDEN:` Hidden text `RCG_STRIKE:` Strike through `RCG_PROTECT:` Protected text `RCG_SUPSCR:` Superscript `RCG_SUBSCR:` Subscript `RCG_ULINED:` Double underlined text `RCG_CAPS:` Capitalize the letters `RCG_SCAPS:` Apply small caps Or, set to 0 to leave this attribute unchanged. Please use the logical OR operator to specify more than one styles.
BOOL set;	Set to TRUE to set the styles specified by the 'style' argument. Set to FALSE to reset the styles specified by the 'style' argument

Description: This text font set by this function is effective for the subsequent RcgInsertText function. The text font can be reset by using the RcgResetTextProp function.

Return Value: This function returns TRUE if successful.

See Also:

RcgTextColor

ASP Interface

This chapter describes the usage of PDF Report Generator within an ASP page. The product includes an additional wrapper DLL called PDGC.DLL which is used to access the converter within an ASP page. Please follow the following steps:

Copy ter18.dll, pdc32.dll, pdg32.dll, rcg32.dll and pdgc.dll to the Windows system directory, or any other directory available at the run-time. Now register pdgc.dll using the regsvr32 system utility. The other dlls do not need registration. Now you are ready to use this product within an ASP page.

Here is an example ASP page which generates a small PDF report and displays the report in the browser:

```
<%@ LANGUAGE = "VBSCRIPT"%>

<%

Option Explicit


Dim obj

Dim DocId

Dim result

Dim PdfBytes


Set obj = Server.CreateObject("pdgc.converter")


rem call obj.SetLicenseInfo("xxxxx-yyyyy-zzzzz",

            "srabnnnnn-n","Your company name")


call obj.SetFlags(1,obj.VAL_PGFLAG_RETURN_MSG_ID)    ' quiet
mode


DocId = obj.NewDocObject()


result=obj.InsertText(DocId,"This is a test line")
```

```
PdfBytes = obj.GenerateReportBufferBytes(DocId)


Set obj = Nothing

%>


<html>

<head>

</head>


<body>

<p> Some text before </p>


<%

      Response.Clear()

      Response.Charset = ""

      Response.ContentType = "application/pdf"


      Response.AddHeader "Content-Disposition",
"inline;filename=" + "test.pdf"


      Response.BinaryWrite(PdfBytes)


      Response.Flush()

      Response.End()


%>


<p> Some text after </p>

</body>
```

```
</html>



<body>

<p> Some text before </p>


<%= sPdf %>


<p> Some text after </p>

</body>

</html>
```

When the above asp file is loaded, IE displays generated pdf code for the report.

The method names used by the pdgc.dll are the same as the functions mentioned in the Application Interface functions. However the 'Pdg' and 'Rpg' prefixes are not used by the pdgc method names. For example, the PdgGenerateReportFile function is named as GenerateReportFile within the pdgc.dll file.

Also, the constants values are prefixed with an 'VAL_' prefix. For example, the constant PGPROP_HYPERLINK becomes VAL_PGPROP_HYPERLINK.