Software License Agreement

PDF Report Generator

for .NET

Version 5

SUB SYSTEMS, INC.

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 $50 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, .NET 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 package and provides a step by step process of incorporating PDF Report Generator into your application.

In This Chapter

Files

The package contains the pdgn.dll, rcn.dll, TESN23.DLL and PDN11.DLL files necessary to incorporate this product into your application.

The package also includes a set of files to construct a demo program. The demo program shows by example the process of linking the DLL to your program.

DLL Demo Files:

The following demo files are included in the c_demo.zip file.

demo.cs	Source code for the demo program
demo.exe	Executable demo program
demo.csproj	The project file to compile the demo.
AssemblyInfo.cs	Assembly information file

Visual Basic Interface and Demo Files:

Form1.vb	vb source file
dmo_vbn.vbproject	The project file for the visual basic demo program.
AssemblyInfo.vb	Assembly information file for the demo program.

License Key

Your license key is e-mailed to you after your order is processed. You would set the license key using the PdgSetLicenseKey static function. This should be preferably done before creating the Pdg object to avoid pop-up nag screens.

Pdg.PdgSetLicenseKey("xxxxx-yyyyy-zzzzz")

Replace the 'xxxxx-yyyyy-zzzzz' by your license key.

Creating a sample document

Please ensure that PDGN.DLL, RCN.DLL, TESN23.DLL and PDN11.DLL files are available in the project directory. Set the reference for PDGN.DLL and RCN.DLL in your project.Now set namespace for the product:

```
// C# example
  using SubSystems.RC;              // C# example
  using SubSystems.PDG;

' VB Example
  Imports SubSystems.RC
  Imports SubSystems.PDG
```

Now set the product license key and create an Rcn type object:

Pdg.PdgSetLicenseKey("xxxxx-yyyyy-zzzzz")

Next, create a Pdg object (pdf generator object) and a Rcn object(report creator object). The Rcn document object is used to create a formatted report. Once the report coding is complete, the Pdg object is used to generate the PDF file for the report.

```
// C# example
  Pdg pdg = new Pdg();
  Rcn doc = pdg.PdgNewDocObject();

' VB example

  dim pd as Pdg
  pd = new Pdg

  dim doc as Rcn
  doc = pd.PdgNewDocObject()
```

The following sample report 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# Example:

```
Pdg pdg = new Pdg();
pdg.InWebServer=true; // should be set only when using the
                      // report generator in an ASP.NET app
                      // to suppress messages

Rcn doc = pdg.PdgNewDocObject();



doc.RcgInitSect();   // begin first section
doc.RcgBeginSectText(); // begin the text for the section
doc.RcgTextFont("Times New Roman", 12, Rcn.RCG_BOLD, true);
doc.RcgInsertText("This is the first line\r\n");
doc.RcgTextFont("Arial", 12, Rcn.RCG_BOLD, false);
doc.RcgParaFlags(Rcn.RCG_CENTER);
doc.RcgInsertText("This is the second paragraph\r\n");

pdg.PdgGenerateReportFile(doc,"test.pdf");
```

Visual Basic Example:

```
Dim pd As Pdg
pd = New Pdg

pd.InWebServer=true ' should be set only when using the
                    ' report generator in an ASP.NET app
                    ' to suppress messages
```

```
Dim doc As Rcn
doc = pd.PdgNewDocObject()

doc.RcgInitSect()    ' begin first section
doc.RcgBeginSectText()   ' begin the text for the(section)
doc.RcgTextFont("Times New Roman", 12, Rcn.RCG_BOLD, True)
doc.RcgInsertText("This is the first line" + vbCrLf)
                  'vbCrLf terminate the(paragraph)
doc.RcgTextFont("Arial", 12, Rcn.RCG_BOLD, False)
doc.RcgParaFlags(Rcn.RCG_CENTER)
doc.RcgInsertText("This is the second paragraph" + vbCrLf)

pd.PdgGenerateReportFile(doc, "test.pdf")
```

Application Interface Classes

The product includes the following two classes:

Pdg: Pdf Generator class. This class is used for two purposes. It is used to create a new 'report creator object'. It is also used to generate the final PDF output using the 'report creator' object. This class is supplied by the pdgn.dll file.

The constants used by the methods in this class must be resolved using the Pdg. modifier. Example:

```
pdg.PdgSetFlags(true,Pdg.RPFLAG_RETURN_MSG_ID)
```

Rcn: Report Creator class: An object for this class is created using the PdgNewDocObject method of a Pdg object. The Rcn class is used to code the body of the report.

The constants used by the methods of this class must be resolved using the Rcn. modifier:

```
doc.RcgParaFlags(Rcn.RCG_CENTER);
```

Please refer to the Getting Started topic for an example of using these two classes to generate a PDF report.

In This Chapter

Pdg Class

This class is used for two purposes. It is used to create a new 'report creator object'. It is also used to generate the final PDF output using the 'report creator' object. This class is supplied by the pdgn.dll file.

The constants used by the methods in this class must be resolved using the Pdg. modifier. Example:

```
pdg.PdgSetFlags(true,Pdg.RPFLAG_RETURN_MSG_ID)
```

Please refer to the Getting Started topic for an example of using this class to generate a PDF report.

In This Chapter

Pdg Class Methods

This chapter includes the public methods offered by the Pdg class.

In This Chapter
PdgGenerateReportFile
PdgGenerateReportString
PdgGetLastMessage
PdgNewDocObject
PdgResetLastMessage
PdgSetFlags
PdgStrToBytes
PdgWriteToFile

 

PdgGenerateReportFile

Generate the PDF file using the Rcn document object.

bool PdgGenerateReportFile(doc, OutFile)

Rcn doc;	// The document object which contains the coded report.
string OutFile;	// Output PDF file containing the report.

Return value: This method returns True when successful.

Examples:

```
pd.PdgGenerateReportFile(doc, "test.pdf")
```

 

PdgGenerateReportString

Generate the PDF in a string using the Rcn document object.

string PdgGenerateReportFile(doc)

Rcn doc;	// The document object which contains the coded report.

Return value: This method returns the generated PDF in a string.

A null value indicates an error condition.

Examples:

```
String PdfStr=pd.PdgGenerateReportString(doc)
```

Please refer to the PdgStrToBytes method for an example of passing the generated pdf report to the ASP.NET Response object.

PdgGetLastMessage

Get the last message.

int PdgGetLastMessage(PdgMessage, DebugMessage);

string PdgMessage;	// Returns the default user message text in English
string 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 RPFLAG_RETURN_MSG_ID flag. This flag is set using the PdgSetFlags method.

 

PdgNewDocObject

Create a new document object for the reporting session.

Rcn PdgNewDocObject()

Description: This function creates the document object for the current reporting session. A document object is needed to call Rcn class methods to create the content of the report.

Return Value: The function returns an Rcn class object when successful. A null value indicates a fail return.

 

PdgResetLastMessage

Reset the last editor message.

bool PdgResetLastMessage()

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.

int PdgSetFlags(set, flags)

bool set;	// TRUE to set the given flags, FALSE to reset the given flags
int flags;	// Flags (bits) to set or reset. Currently, the following flag values are available:

RPFLAG_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.

PdgStrToBytes

Convert a pdf string to a byte array.

byte[] PdgStrToBytes(PdfString)

String PdfString; // Input string containing PDF text.

Return value: This function returns a byte array from the given string. This is a preferred method of converting a pdf string to a byte array because it returns the raw bytes without employing an encoding method.

A null return values indicates an error.

Example:

```
Response.Clear();

Response.Charset = "";

Response.ContentType = "application/pdf";


string strFileName = "test" + ".pdf";

Response.AddHeader("Content-Disposition",

                    "inline;filename=" + strFileName);


Pdg pd = new Pdg();

pd.InWebServer=true; // should be set only when using the

                     // report generator in an ASP.NET app

                     // to suppress messages


Rcn doc = pd.PdgNewDocObject();



doc.RcgInitSect();   // begin first section

doc.RcgBeginSectText(); // begin the text for the section

doc.RcgTextFont("Times New Roman", 12, Rcn.RCG_BOLD, true);

doc.RcgInsertText("This is the first line\r\n");

doc.RcgTextFont("Arial", 12, Rcn.RCG_BOLD, false);
```

```
doc.RcgParaFlags(Rcn.RCG_CENTER);

doc.RcgInsertText("This is the second paragraph\r\n");


string PdfString=pd.PdgGenerateReportString(doc);


Response.BinaryWrite(pd.PdgStrToBytes(PdfString));


Response.Flush();

Response.Close();

Response.End();
```

PdgWriteToFile

Write a pdf string to a disk file.

bool PdgWriteToFile(FileName, PdfString)

string FileName; // Output file.

string PdfString; // Pdf string to be written to the disk file

Return value: This function returns TRUE when successful.

Pdg Properties

The control properties can be before the conversion to affect the pdf output. The control supports the following properties:

InWebServer

This property should be set to True when this control is used in a web server. When this property is set to True, the control suppress the display of any dialog and message boxes.

ProjectFolder

Set this property to the folder containing your project, such as c:\inetpub\wwwroot\MyProject. This information helps the converter locate the images which use relative path. It is also used for creating any temporary files.

Author

Set the author name for the PDF document.

Bookmark

Set to true to convert the rtf table-of-content to PDF bookmark. The default value is true.

CreDate

Set the document creation date. The date is specified in a text string.

Hyperlink

Set to true to translate rtf hyperlink fields to pdf hyperlinks. The default value is true.

Keywords

Set the keywords for the PDF document.

LicenseKey

Set the product license key for the product. *Your license key is e-mailed to you after your order is processed.*

ModDate

Set the document modification date. The date is specified in a text string.

Producer

Set the producer description for the PDF document.

Subject

Set the subject description for the PDF document.

Title

Set the title for the PDF document

CompressText

Set to true to compress the text stream in the PDF output.

PermFlags

Use this flag to specify the permissions granted when the PDF document is being viewed or manipulated without using the owner password. You can use one or more of the following flags using the OR operator:

pc.PERM_PRINT	Allow printing operation
pc.PERM_COPY	Allow copying operation
pc.PERM_MOD	Allow document modification

OwnerPassword

Optional document owner password.

When either an owner or a user password is specified, the PDF document is written out using Adobe standard encryption mechanism.

An owner password in the PDF document requires a PDF editor to prompt the user for the owner password and allow PDF modification only when the supplied owner password matches the encrypted owner password found in the file.

UserPassword

Optional user password

When either an owner or a user password is specified, the PDF document is written out using Adobe standard encryption mechanism.

A user password in the PDF document requires a PDF viewer to prompt the user for the user password and allow PDF display only when the supplied user password matches the encrypted user password (or owner password) found in the file.

EmbedFonts

Normally the converter only embeds non-standard fonts in the PDF file. This flag would instruct the converter to embed all fonts.

KeepRowTogether

Keep the table row on one page.

Rcn Class

The Rcn class is used to code the body of the report.

An object for this class is created using the PdgNewDocObject method of a Pdg object.

```
doc = pdg.PdgNewDocObject()
```

The constants used by the methods of this class must be resolved using the Rcn. modifier:

```
doc.RcgParaFlags(Rcn.RCG_CENTER);
```

Please refer to the Getting Started topic for an example of using this class to build a report.

In This Chapter

RcgParaIndent
RcgParaShadeColor
RcgParaSpace
RcgParaStyleId
RcgParaTabStop
RcgResetLastMessage
RcgResetTextProp
RcgResetParaProp
RcgSectInfo
RcgSetFlags
RcgTextColor
RcgTextFont

RcgBeginCellText

Begin the text for a cell.

bool RcgBeginCellText()

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(PageRelative,x, y,width,height)

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(name,IsDest)

string 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(type)

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()

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(ParaStyle,StyleId,name)

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.
string 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()

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(indent,height,just,IsHdr,CellMargin)

bool RcgBeginTableRow2(indent,height,just,IsHdr,LeftMargin, RightMargin, TopMargin, BotMargin)

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 valid for the RcgBeginTableRow method only.
int LeftMargin;	The distance of the text in the cell to the left edge of the cell. Set to 0 for default. This parameter is valid for the RcgBeginTableRow2 method only.
int RightMargin;	The distance of the text in the cell to the right edge of the cell. Set to 0 for default. This parameter is valid for the RcgBeginTableRow2 method only.
int TopMargin;	The distance of the text in the cell to the top edge of the cell. Set to 0 for default. This parameter is valid for the RcgBeginTableRow2 method only.
int BotMargin;	The distance of the text in the cell to the bottom edge of the cell. Set to 0 for default. This parameter is valid for the RcgBeginTableRow2 method only.

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

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.
Color FillColor;	The background color of the text box. Set to Color.White for default.
Color BorderColor;	The color of the border. Set to Color.Black 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(id)

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(PageRelative,x1,y1,x2,y2,ZOrder,LineColor, LineWidth)

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.
Color LineColor;	The color of the line object. Set to Color.Black 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(PageRelative,x,y,width,height,ZOrder,FillColor,
BorderColor,BorderWidth,Xparent)

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.
Color FillColor;	The background color of the rectangle object. Set to Color.White for default.
Color BorderColor;	The color of the border. Set to Color.Black 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()

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()

Return Value: This function returns True if successful.

RcgEndGroup

End the current RTF group.

bool RcgEndGroup()

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()

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()

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()

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()

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()

Return Value: This function returns True if successful.

See Also:
RcgBeginTextBox

RcgGetLastMessage

Get the last message.

int RcgGetLastMessage(RcgMessage, DebugMessage);

string RcgMessage;	// Returns the default user message text in English
string 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.

See Also

RcgResetLastMessage

 

RcgInitSect

Initialize a section.

bool RcgInitSect()

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(name, start)

string 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(type)

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(name,UseParam,param)

string 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(marker, text)

string marker;	The footnote number text. Set this parameter to NULL to specify auto footnote numbering.
string 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.

RcgInsertHyperlink

Insert a link to an external document or an internal bookmark location.

bool RcgInsertHyperlink(LinkText, IsLocal, target)

string LinkText;	link text
bool IsLocal;	Set to true to insert a link to an internal bookmark location. Set to false to link to a web location or an external document.
string target;	Set the target parameter to an internal bookmark name when the IsLocal parameter is set to true. The document must include a bookmark using the same name as the target. You can create a bookmark location using the RcgInsertBookmark method. Set the target to an web url or an external document name when the IsLocal parameter set to false.

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

Return Value: This function returns True if successful.

RcgInsertImage

Insert an Image object.

bool RcgInsertImage(image,width,height)

Image image;	The image object to insert. This method does not support metafile image objects.
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.

Description: The image is written out to the rtf file using the PNG format.

Return Value: This function returns True if successful.

RcgInsertMergeField

Insert a mail-merge field.

bool RcgInsertMergeField(name,data)

string name; field name

string 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(InsertPageNo)

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(file,width,height,linked)

string 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, PNG, JPG, GIF, EMF 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(code)

string 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.

RcgInsertTocField

Insert Table-of-Content field.

bool RcgInsertTocField(FirstLevel,LastLevel)

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

RcgInsertText

Insert text into the document.

bool RcgInsertText(text)

string 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(LeftMargin,RightMargin,TopMargin,BotMargin, HdrMargin,FtrMargin)

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

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

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.
Color BackColor;	Background color for the cell. Set to Color.White 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.
Color BorderColor;	Cell border color.
int 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(width,height,landscape)

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(IsBullet,type,level,BefText,AftText,flags)

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.
string BefText;	The text before the paragraph number. Set to NULL for default.
string AftText;	The text after the paragraph number. Set to NULL for default.
int 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(flags)

int 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(LeftIndent,RightIndent,FirstIndent)

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(shade,BkColor)

int shade;	The paragraph shading in percentage. Set to 0 to set paragraph background color instead.
Color 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(SpaceBef,SpaceAft,SpaceBet)

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(id)

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(type,pos,flags)

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.	
int 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

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()

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()

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(columns,ColSpace,NewPage,FirstPageNo)

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.

int RcgSetFlags(set, flags)

bool set;	// True to set the given flags, False to reset the given flags
int 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(CurColor,foreground)

Color 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(typeface,PointSize,style,set)

string 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.
int 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