PDF REPORT WRITER

FyTek, Inc.

Web site: https://www.fytek.com

FyTek's PDF Report Writer

Trademarks

FyTek, FyTek PDF Report Writer and the FyTek logo are registered trademarks or trademarks of FyTek Incorporated in the United States and/or other countries. Acrobat, Adobe, Adobe PDF and Adobe Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product names, logos, designs, titles, words or phrases mentioned within this publication may be trademarks, servicemarks, or tradenames of FyTek, Inc. or other entities and may be registered in certain jurisdictions including internationally.

FyTek Disclaimer

FYTEK, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE. Copyright © 2000-2023 FyTek, Inc. All rights reserved. This manual may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of FyTek, Inc.

This guide may contain links to third-party websites that are not under the control of FyTek, and FyTek is not responsible for the content on any linked site. If you access a third-party website mentioned in this guide, then you do so at your own risk. FyTek provides these links only as a convenience, and the inclusion of the link does not imply that FyTek endorses or accepts any responsibility for the content on those third-party sites.

Acknowledgments

Software Development: Mike Bernardo
Documentation: Mike Bernardo

FyTek, Inc.
P.O. Box 71093
Madison Heights, MI 48071

FyTek, Inc.

Digitally signed by FyTek, Inc.
Reason: Attestation to the accuracy and integrity of this document
Location: FyTek, Inc., Michigan, US
Sunday June 11, 2023 06:17:46 pm
Cert Serial: 08 FD B4 33 16 3B A3 A4 01 B1 8A F2 0E E6 DE 13

Introduction

PDF Report Writer has many features to make creating reports in PDF (Portable Document Format) as easy as using HTML. This document describes the commands available and how to go about arranging them to produce a report in PDF.

The main purpose of PDF Report Writer has always been to create great looking PDF files while allowing flexibility in the layout and ease of use. However, you may also use PDF Report Writer to create XPS documents that are based on Microsoft's specifications and look basically identical to their PDF counterpart.

In addition Excel, CSV, HTML and RTF (Rich Text Format) are still other output options you have when using the product; however, those formats do not retain the look and feel of the PDF or XPS layouts.

Getting Started

Different methods of building a PDF are used depending on the version of Report Writer you are working with. The executable version (for DOS or Linux) takes as input a plain text or ASCII file and produces a PDF. The input file must contain valid Report Writer commands as defined later in this document. Alternatively, you may pass the commands in via the environment variable PDFCMD. This can be useful in Linux so you don't have to create an input file when creating PDFs for viewing on the web.

The DLL wrapper can similarly take a plain text file from disk or read commands into memory as passed in from languages such as Visual Basic or PHP. The method setPDFCmd is used to pass in one or multiple commands. Once all the commands are passed in and any other settings are made the buildReport method takes care of creating the PDF. The PDF can be created as a file on disk or as a character stream for display in a browser. Data may be included within the layout or use external XML or JSON formatted data with a layout template.

Sample input files are included with the software so you can see how to design your input. These files have a .frw extension and start with the word "sample". Note the data for your report may come from an outside application where you generate the final commands for Report Writer or you may include SQL queries in the report input itself. You will need to use a supported database driver based on the operating system you are using.

PDF Report Writer is available in different configurations. The chart below shows the features of each.

Version	Pop-Up Message (message "You are using FyTek's PDF Report Writer...")	Server/Multi-User (for use on a server or in a web environment)	All Features (charting, background PDFs, XML and XPS)
PDF Report Writer Single User	✓		✓
PDF Report Writer SE* (Server Edition)		✓	
PDF Report Writer EE (Enterprise Edition)		✓	✓
PDF Report Writer Software Subscription		✓	✓

* SE version does not include charting, ability to use background PDFs or XML.

Contact FyTek at sales@fytek.com for details on what version is right for you.

Reports

A report as defined by this document is a collection of pages in a single PDF file. Each report page may have up to three sections - a header, body and footer. The header and footer are repeated on each page for as many pages that the body of the report takes. The contents of the body will wrap across as many pages as necessary to fit all of it. The software will automatically calculate the number of pages needed and create the minimum number necessary to hold all of the content.

The layout when using headers, footers and a body is to issue the PAGE command followed by the HEADER and/or FOOTER then the BODY. The header and footer must appear before the body in order to determine how much space the body has. Remember to include the BODY tag when using a header or footer or you'll likely end up with body text overlaying your header or footer.

The following sections describe the options available for the Windows/Linux versions then the methods for the DLL. All versions have the same functionality (with the exception of some Windows API specific functions missing from Linux) with regards to the Report Writer command language. See the Executable Examples section for information on calling the executable from languages such as Perl or PHP. See the Linux Executable section for information on Linux.

Using the Executable

PDF Report Writer can be setup to run either stand-alone or in server mode. Under Windows, PDF Report Writer can be setup in server mode as a Windows service allowing users to build their reports centrally off of one computer. Unix users can setup the server in a similar fashion by starting the server as a background process. Requests are sent to the program via TCP/IP when running in server mode. See the Client-Server section for detailed information on this type of setup.

The program pdfrw.exe (or pdfrw on Linux/Unix) is used to create a PDF from commands stored in a file or passed in from the DLL. The 64-bit version is pdfrw64.exe for Windows and pdfrw64 for Linux. If you are running PDF Report Writer in server mode, clients can use the program pdfrw_tcp or pdfrw_gui_tcp (or their 64-bit equivalents) in addition to the DLL. Execute the program by running:

pdfrw.exe (or pdfrw64.exe) *filein.frw fileout.pdf*

where "filein.frw" is the name of your input file and "fileout.pdf" is the PDF output. The file "filein.frw" will contain a set of tags or commands that describe the format of the resulting PDF. You must make sure your tags are opened and closed appropriately or you may not get a valid PDF. For example, for each <TD> tag you should have a closing </TD>. All of the tags and their options are described in the Commands section of this document. If you get an error it will most likely be that a tag is misplaced or missing. In that case check your report file carefully and make whatever adjustments are necessary.

You may also specify a web page in the form http://www.mysite.com/page.html though keep in mind Report Writer is not intended as an HTML to PDF converter. The web page option should only be used on pages with a layout suited for Report Writer (no complex graphical layout, no tables within tables that won't fit on a page, etc). You may also use a file that contains Report Writer commands rather than HTML. A file that begins with the <PDF> tag will be assumed to contain Report Writer commands and not HTML. Be sure to specify a page and not just the site. Specify the proxy, if necessary, as a DOS environment variable. The statement would be something like:
set http_proxy="http://proxy:8080"

A file type extension of .frw is set up when the product is installed. This will allow you to launch Report Writer on a .frw type file from Explorer or by simply typing in the name of the file on the command line. Running pdfrw.exe with no parameters will bring up a file open dialog box and allow you select an input file.

You may use a single dash '-' as the input file to specify input is from standard input (typically the keyboard). You can use this feature to pipe in data from a file or other source. For some situations (such as passing data from an environment variable) you may want to bypass entering an input file but still use 'stdout' as the output. Use -none as the input file name in that case. This option must be the first one passed in as is required for the input file name.

Use 'stdout' as the output file name to send the output to standard output (typically the console) rather than a file. This allows you to pipe the output elsewhere or use the executable in a web environment when you don't want an output file created.

You may also set to the string 'genfile'. Report Writer will generate a unique name for the output PDF in the temporary directory. You can use the option -outdir to assign a different directory to use. See the DELETEPDF tag if you want to delete the PDF after it's been built (for example, when you just want to email it).

All options may be passed in any order except for the input and output files. Those must be the first and second parameters when used. You may run something like "pdfrw.exe sample.frw -open" however and the program will understand you want to open the output PDF, not call it "-open". The output file will automatically be named sample.pdf in this case.

Other options you can pass to pdfrw are:

-aes 128|256

Sets AES encryption method. Pass 128 for 128-bit encryption or 256 for 256-bit encryption. Files encrypted with AES 128-bit encryption can only be opened with Acrobat or Adobe Reader 7.0 or above. Files encrypted with AES 256-bit encryption can only be opened with Acrobat or Adobe Reader 9.0 or above.

-allowbreaks

Allows use of line breaks within input tags. Line breaks are only allowed between options and not within an option (including the option word, and equal sign and value if the option takes a value). For example:

```
<TD WIDTH=30 NOWRAP>
```

can be written as:

```
<TD
WIDTH=30
NOWRAP>
```

-allowperl

Allows parsing of Perl code in the input (layout) file. See the XML Data Merge section for information on using the PERL tag.

-ascii85

Apply ASCII85 encoding scheme to PDF. Useful if emailing PDF to certain providers which may not handle the PDF attachment properly.

-b

Run the program in background mode so no dialog boxes are displayed for errors. Use the -statfile option to capture results if necessary.

-b1 *text*

Sets the text for button 1 (default is "Cancel").

-b2 *text*

Sets the text for button 2 (default is "Break on next page").

-b3 *text*

Sets the text for button 3 (default is "Pause").

-b4 *text*

Sets the text for button 4 (default is "Resume").

-buildalert *path-file*

Optional. The path and name of a file to create when finished processing. The contents of the file will contain the number of PDFs created followed by a line feed.

-buildlog *path-file*
Optional. The path and name of a file to log output file names and number of pages, tab separated. This file is appened to if it exists so you may want to delete the file before running PDF Report Writer.

-cancelonly
Displays only the "Cancel" button in the progress box.

-clean
Automatically deletes the input file when finished building the PDF.

-comp15
Uses a compression algorithm compatible with PDF 1.5 (Acrobat 6.0). PDFs with this form of compression can be viewed only with Acrobat or Reader version 6 or higher. The reduction in size is based on the number and type of objects in the PDF but in general is around 10-20%. Not all PDFs will be reduced by the same percentage factor.

-copies *number*
Number of copies to print when using the -print or -printer commands. Default is 1.

-copycsv *filename.csv*
Copies the CSV file to the specified location and file name once it's built.

-copyhtml *filename.html*
Copies the HTML file to the specified location and file name once it's built.

-copypdf *filename.pdf*
Copies the PDF to the specified location and file name once it's built.

-copyrtf *filename.rtf*
Copies the RTF file to the specified location and file name once it's built.

-copyxls *filename.xls*
Copies the Excel file to the specified location and file name once it's built.

-copyxps *filename.xps*
Copies the XPS file to the specified location and file name once it's built.

-csv *filename.csv*
Creates a CSV (comma separated values) file using the filename specified. Must use the EXCEL tag with this option.

-csvcr
Adds a carriage-return (ASCII 13) to the end of each line before the line-feed character. Mostly for DOS when creating a txt file to be opened by Notepad.

-csvdelim *character(s)*
Sets the delimiter for the CSV file. Pass in the character(s) you want to use. The default is a comma. Use either this option or csvdelimnum.

-csvdelimnum *number*
Sets the delimiter for the CSV file. Pass in the ASCII number code of the character you want to use. The default is 44 which is the comma. Use either this option or csvdelim.

-csvindelim *character*
Treats the input file as a CSV file rather than a file starting with the <PDF> tag. The file is assumed to contain a set of values, one row per line, with the specified delimiter. The data is placed in a table structure and is used to generate the PDF.

-csvinhdrrows *number*
The number of header rows in the input CSV file. These rows are placed in TH (repeat) table rows for reprinting at the top of each page.

-csvinheader *text*
The title for the input CSV file. Use <BR> for a new line. You may also use FONT and other formatting tags. This will print in the header section of the report.

-csvmail
Opens the user's email program to a composition window with the newly created CSV file attached. May not work with all email programs. Must use the -csv option to create a CSV file for this option to work.

-csvopen

Opens the CSV file with the default viewer associated with CSV files.

-cwd *text*

Changes the working directory to the specified directory.

-data *filename.dat*

(Not available in SE version)
Sets the data file to use with the input (layout) file. May be a file on the system or web address of a file. See the XML Data Merge section for information on merging a layout file with a data file.

-datacheck *filename.frw*

Exports a file similar to the input but formatted to tab indent the DATASET tags for review. Also adds the ID to closing DATASET tags and a PATH attribute that shows the current nested XML level. Checks for any opening/closing TABLE issues in relation to DATASET tags.

-dataenc *text*

(Not available in SE version)
Optional. Sets the format for the -data file. Data is assumed to be in UTF-8 format which is converted to ISO-8859-1. Pass "utf8" for UTF-8 or "iso" for ISO-8859-1 or plain ASCII.

-datakeyattr *text*

(Not available in SE version)
A comma separated list of attributes to break out from standard XML files. See the XML Data Merge section for information on merging a layout file with a data file.

-dataout *filename.dat*

(Not available in SE version)
Standard XML data files are converted to data files used by Report Writer. This option exports the converted XML file so you can determine if any changes are needed. See the XML Data Merge section for information on merging a layout file with a data file.

-debug *debugfile*

Used to produce a flat file of commands being passed to the program for debugging. Full path and name of the output log file. Useful for debugging XML data merges.

-deletepdf *[pause,[timeout]]*

Deletes the output PDF once the program is finished. This is useful when emailing the PDF and you don't want to retain a copy of the PDF that was sent. You can also set this with the DELETEPDF tag. The pause and timeout (optional parameters) are described in the DELETEPDF tag section.

-delim *character*

Delimiter to use if you want to create a txt or csv file instead of PDF. A text file is created (rather than PDF) if your output file ends with txt or csv. This is mostly for debugging or quick retrieval of data without going through the PDF conversion process.

-dsnencrypt *text*

Used to encrypt a DSN file to provide added security. In this case, pass a plain text file with the DSN information as the input file and the name of the encrypted file you wish to use as the output file. PDF Report Writer will simply encrypt the data using the password provided. The output file can then be used with the parameter -dsnfile when running PDF Report Writer in server mode. The -dsnpwd option, again as part of the server startup, should be set the same as what was used for -dsnencrypt. See the DSN Layout for details on the DSN file structure.

-e *logfile*

Used to produce an error log file. Use this option during development and debugging. No error checking is done if this is not used. The error file is not created if no errors are found while parsing the commands. If any errors are found, they are placed in this file and the PDF file is not created. See the error page for a list of errors that are checked for.

-e128

Sets 128-bit encryption method. Files encrypted with 128-bit encryption can only be opened with Acrobat or Adobe Reader 5.0 or above. The default encryption is 40-bit which works with Acrobat and Adobe Reader 4.0 and above.

-f *width,height,point-size*

Used to convert plain text file to PDF. This option treats the input as plain text with no formatting. The height, width and point-size are optional. The default width is 8.5, default height is 11 and default point-size is 10. To specify landscape with a 9.5 point font, use -f 11,8.5,9.5.

-force *text*

Turns off the dialog box prompting to overwrite the output file if it exists.

-guioff

Suppresses the dialog window that shows the current build progress.

-html *filename.html*

Creates an unformatteed HTML file using the filename specified. Must use the EXCEL tag with this option. Note you may use filename.xls and have Excel open with the -htmlopen option. Newer versions of Excel will open and convert HTML files automatically.

-htmlconv

Use when passing an HTML formatted file as input. Report Writer will perform some pre-processing on the file in an attempt to better format it. Only simple (text & single level tables) should be used in this case.

-htmlopen

Opens the HTML file with the default viewer associated with files of the file extension given.

-ini *inifile*

Configuration file containing parameters to pass in. The commands may be all on one line or on multiple lines. Any of the options in this section may be used. You can also set an environment variable called RWCFG (the name is case sensitive) and have it point to an initialization file.

-insize *filename.siz*

Specifies the input table sizing file to use. The PDF being built must match the layout the sizing file was created for. See the Build Options section for more information.

-lowpriority

Sets the priority of the program lower allowing other applications to run faster.

-mail

Opens the user's email program to a composition window with the newly created PDF attached. May not work with all email programs.

-mailscr *file*

For Unix/Linux systems where -mail is not available. Specify a script that will receive as a parameter the output PDF file name. Create a script for your operating system that will be used to bring up an email window with an attached PDF.

-movecsv *filename.csv*

Moves the CSV file to the specified location and file name once it's built.

-movehtml *filename.html*

Moves the HTML file to the specified location and file name once it's built.

-movepdf *filename.pdf*

Moves the PDF to the specified location and file name once it's built.

-movertf *filename.rtf*

Moves the RTF file to the specified location and file name once it's built.

-movexls *filename.xls*
Moves the Excel file to the specified location and file name once it's built.

-movexps *filename.xls*
Moves the XPS file to the specified location and file name once it's built.

-named *filename.dat*
Used to add named page templates to the output PDF. This is mostly for FDF forms. Adds in a template name of "page#" where # is replaced with the corresponding PDF page number. A file name is passed in which will contain the total number of pages in the PDF along with the template names used. Use a value of 1 for the file name to not write this information to a file but still place the template names in the PDF.

-noannote
Disables add/change of form fields or annotations. Use with the -o or -u option.

-nochange
Disables changes to the document. Use with the -o or -u option.

-nocopy
Disables copying of text and/or graphics from the document. Use with the -o or -u option.

-nopdf
Skips creating a PDF. Useful if you are only generating some other output, like CSV, HTML or EXCEL. Do not use when generating RTF or XPS however.

-noprint

Disables printing of the document. Use with the -o or -u option. To create a PDF with both printing and copying disabled for the user you would run something similar to:

```
pdfrw.exe filein.frw
fileout.pdf -o abc123
-u xyz -noprint -nocopy
```

The file could only be opened by someone who knows one of the two passwords (abc123 or xyz). Using a password of abc123 gives full access while using the password of xyz does not allow printing or copying of text.

-nospin

Turns off the animation in the dialog box.

-np

Turn off the box that shows how far along the program is in building the pdf.

-o *password*

Sets the owner password for the PDF. If not specified but the user password is, this is set to the user password. Also, when not specified, the owner has only the rights granted when the document was created. So for example, if -noprint was specified, then it is impossible for the owner to print the document.

-ooserver *server:port*

The server and port that OpenOffice server is running on. For when you have an OpenOffice server running to convert Excel/Word/etc. to PDF for use by PDF Report Writer. Pass the server and port number. For example, -ooserver "myooserver:2050". See the PDFPAGE command for details on these file types.

-open

Automatically opens Acrobat and loads the newly created PDF.
_000041_

-openscr *file*

For Unix/Linux systems where -open is not available. Specify a script that will receive as a parameter the output PDF file name. Create a script for your operating system that will be used to open PDFs.

-opt

Optimize the output PDF for fast web viewing. Note this typically increases the size of the output by a few hundred bytes or so. The PDF is optimized for viewing on the web as opposed to shrinking the physical size. Additionally, you must create the PDF to a file rather than stream the output to the browser. The setting "fast web view" will be set to yes for optimized PDFs when you open in Reader and check the properties. This means the first page of the PDF is sent to the user and made viewable while the rest of the pages continue to download in the background.

-opt15

Optimize the output PDF for fast web viewing and use higher compression (same as setting -opt and -comp15). The PDF is optimized for viewing on the web and further compressed for a smaller file size. Additionally, you must create the PDF to a file rather than stream the output to the browser. The setting "fast web view" will be set to yes for optimized PDFs when you open in Reader and check the properties. This means the first page of the PDF is sent to the user and made viewable while the rest of the pages continue to download in the background.

-origtablepct	Use for reports designed with an older version of Report Writer (prior to 2012) that use the WIDTH option on TD cells in tables and the TABLE tag does not contain a WIDTH value. Only needed if your report is not generating correctly with the current version.
-outdir *directory*	Optional when using "genfile" as the output file. This is the directory to create the output PDF in. The default is to use the current users' temporary directory.
-outsize *filename.siz*	Creates a table sizing file for use with future builds of similar PDFs. See the Build Options section for more information.
-pause *value[,seconds]*	Used to prevent the program from taking up 100% of the CPU. Use this option if you're running several instances of the program or other programs at the same time. The value should be set somewhere between 1 and 20 with lower numbers giving better CPU utilization but longer PDF build times. You may optionally set the number of seconds to pause with the default (and minimum value) being .001 seconds. Do not place a space before or after the comma when setting the number of seconds.
-pbm *message*	Sets the message of the progress dialog box.
-pbt *title*	Sets the title of the progress dialog box.
-pdf *bkgfile.pdf*	*(Not available in SE version)* Full path and name of the PDF to pull pages from as backgrounds. See the backgrounds section for more information.
-pdfver	The version number to assign the PDF. The default is 1.4 (Acrobat 5.x).

-perldbi	May need to specify on some Unix systems for processing of CSV files using SQL.
-pidfile *pidfile*	Creates a file which contains the process ID of the application. The file is removed once the PDF is build or is cancelled.
-pidmsg *hWnd,msg*	Sends a message, using Windows SendMessage function, to the window handle specified. Pass two numbers separated by a comma with no space before or after the comma. The first number is the window handle to send the message to and the second is the message id number to use. See the Windows API for information on SendMessage. The value of wParam will be the process ID.
-plain *character*	Sets plain mode for tables. You must specify a character when setting on the command line. See the PLAIN tag for more information.
-print	Automatically prints the newly created PDF to the default printer. Must have Acrobat or Adobe Reader installed.
-printcopies n	Works with Acrobat or Reader 8.0 or higher. The number of copies to be printed when the print dialog is opened for this file. Supported values are the integers 2 through 5. Values outside this range are ignored.
-printdlg	Brings up the Acrobat print dialog box and allows printer selection. This only works when the user has Acrobat or Adobe Reader associated with PDFs on their machine. Otherwise the user's viewer is opened with the document and they will need to print from there.

-printduplex "text"

Works with Acrobat or Reader 8.0 or higher. The paper handling option to use when printing the file from the print dialog. The following values are valid:
Simplex Print single-sided
DuplexFlipShortEdge Duplex and flip on the short edge of the sheet
DuplexFlipLongEdge Duplex and flip on the long edge of the sheet

-printer *printer* *device* **port**

Used to print the PDF to the specified printer. There is no print dialog box in this case. This option takes three parameters: printer, device and port. You may pass in just the printer and leave device and port blank to use the default settings for the printer. For example:
-printer "Accounting Printer" "HP LaserJet 5" "lpt1:"
or
-printer "Shipping Printer"

You may also use the printer port as the first parameter and leave the last two off if you are using a network priner or don't know the printer name. For example:
-printer "\\server\printer"

-printerlist *file*

Used to generate a list of printers available on the system. This can be used to verify what printers the program finds and what they are called. The list generated is tab separated and includes the printer name, device name and port. Use any of the printer names in the file with the -printer option. This option is only available under Windows systems. Use this option by itself as the program will exit after generating the list.

-printpagerange "from,to,..."

Works with Acrobat or Reader 8.0 or higher. The page numbers used to initialize the print dialog box when the file is printed. The first page of the PDF file is denoted by 1. Each pair consists of the first and last pages in the sub-range. An odd number of integers causes this entry to be ignored. Negative numbers cause the entire array to be ignored.

-printpicktray

Works with Acrobat or Reader 8.0 or higher. Specifies the PDF page size is used to select the input paper tray. This setting influences only the preset values used to populate the print dialog presented by a PDF viewer application. If used, the check box in the print dialog associated with input paper tray is checked.

-printscale "text"

Works with Acrobat or Reader 7.0 or higher. Valid values are None, which indicates that the print dialog should reflect no page scaling, and AppDefault, which indicates that applications should use the current print scaling.

-printscr *file*

For Unix/Linux systems where -print is not available. Specify a script that will receive as a parameter the output PDF file name. Create a script for your operating system that will be used to print PDFs.

-q

Quick build - performs fewer passes through the file but some features, such as links, will not be available. See the Build Options section for more information. This option builds the PDF in memory first.

-q2

Quick build method 2 - performs fewer passes through the file but some features, such as total number of pages, will not be available. See the Build Options section for more information. This option builds the PDF to disk (rather than memory first) when an output file is used. It's best to use this option over -q unless you are using the &totpage variable.

-removeflds

Removes any form fields from background PDFs.

-rtf *filename.rtf*

Creates an RTF (Rich Text Format) file using the filename specified. It is recommended that you fully qualify the path or the file will end up in your default document area (i.e. "My Documents" directory). Also see the META tag and RTFIMAGES tag for RTF options.

-rtfopen

Opens the newly created RTF file.

-rwquerystr *text*

Default values to pass in for dynamic processing. This overrides any settings from the environment variable RWQUERY_STRING. Pass name/value pairs with an ampersand in-between. For example, "$reg='A','B'&$drink=ice%20water" will set the variable $reg to "'A','B'" and $drink to "ice water". You can then use $reg and $drink as part of dynamic SQL queries or other logic. See the SQL section for details on queries and loop processing.

-s

Include subdirectories when batch converting. When doing a batch conversion, do not specify an in or out file. Just specify the base directory instead. For example, `pdfrw.exe c:\myfiles\ -s -t frw` will convert all files with a .frw extension in c:\myfiles and all of its subdirectories.

-server

Starts up PDF Report Writer in server mode. PDF Report Writer will run as a separate process and take commands from the TCP/IP port it is connected to. The Client-Server section describes this operation in more detail.

-signbgcolor *text*

Optional. The background color for the signature field. See the Digital Signature section for details.

-signimg *text*

Optional. The path and name of an image to use for the signature. Set this option to "none" to not place any image in the signature field. See the Digital Signature section for details.

-signkeepratio

Optional. Keep the image x/y scaling ratio when using an image with a signature field. See the Digital Signature section for details.

-signname *text*

The name of the signature field in the PDF to sign. This must be an INPUT field of type SIGNATURE as part of the input. For example, "sig1". See the Digital Signature section for details.

-signpemfile *text*

The path and name of the signing certificate. For example, "c:\keys\mykey.pem".
See the Digital Signature section for details.

-signpkfile *text*

The path and name of the private key file. For example, "c:\keys\mykey_pk.pem".
See the Digital Signature section for details.

-signrsn *text*

Optional. The reason for signing the document. Default is "Attestation to the accuracy and integrity of this document". See the Digital Signature section for details.

-signsize *text*

Optional. The fontsize for the text of the signature (0 for no text). Default is 12. See the Digital Signature section for details.

-signssl *text*

The path and file name of the OpenSSL program. For example, "c:\openssl\bin\openssl.exe". See the Digital Signature section for details.

-signtimestamp *url*

The path and name of your timestamp server if you want to include an embedded timestamp on the signature. Use "1" to use the default of https://freetsa.org/tsr. Pass your user name and password if your timestamp server requires it like this: "myuserid:mypassword@https://mytimestamp.org". Use the -signts option with -signtimestamp.

-signts *file*

The path and name of the ts.exe (or just ts for Linux) executable that is used to perform the timestamping function. This is a separate download available here. Use this program for both Windows and Linux.

-sqlcmds

Load the needed SQL libraries on Unix. Because of differences among Unix systems, the libraries may not be compatible so they are not loaded unless this option is used.

-sqldb *text*

The database schema or driver information. See the Database Connection section for details.

-sqldriver *text*

The data source. This is a case-sensitive string. Available for Windows/Linux only. Valid values are:
CSV (or CSVPP)
Fixed
Tab
Pipe
Oracle (32-bit only)
mysql (or mysqlPP)
ODBC
XML

CSV may give slightly better performance over CSVPP on Windows systems.
CSVPP, Fixed, Tab, and Pipe do not support all SQL options.

-sqldsn *text*

The DSN name from the Report Writer DSN configuration file to use. DSN names are setup when running Report Writer in server mode and using the -dsnfile option. You do not need to pass other SQL connection options in this case as they will be handled on the server.

-sqlpwd *text*

The password for the user when making a database connection. This may also be specified on the QUERY tag.

-sqluser *text*

User name to use when making a database connection. This may also be specified on the QUERY tag.

-statfile *statfile*

Used to specify a status file to create once the PDF is built. This file will contain the word OK followed by a line feed if the PDF finished building. It will contain CANCEL followed by a line feed if the PDF build was cancelled. It will contain ERROR followed by a line feed if the -e option was used and errors were found causing the build of the PDF to stop. Any other error that occurs, like not being able to open the output file for the PDF, will be placed in this file. Use this feature if you need to monitor the progress of the PDF build from another program.

-t *type*

File extensions to look for when batch converting. Enter "frw" to convert all files whose extension is .frw.

-tempfile *filename.tmp*

Specify the file to use - DO NOT use this without reading the Build Options section. The file is deleted after the PDF is built. Used to perform fastest build - makes only one pass through the input file, however table column sizes must be specified.

-u *password*

Sets the user password for the PDF. Setting this option will require a password to be entered in order to open the PDF.

-untaint *number*

Untaints file names. Use this on Unix systems if you get errors about "insecure dependency while running setgid" or want to restrict what file names may be used. The parameter takes a number from 1 to 3. A value of 1 allows the least characters and 3 the most. For example, if you don't want to allow files from other directories, use 1. Use 2 if you do allow file names that contain slashes (so directories can be used) or 3 for any character in a file name (such as ! or $).

1 - Only letters and numbers as well as -, _ and @ will be allowed
2 - Same as 1 except also allow \, / and :
3 - Allow all characters

Note this applies to all files - both input and output.

-xls *filename.xls*

Creates an Excel file using the filename specified. It is recommended that you fully qualify the path or the file will end up in your default document area (i.e. "My Documents" directory). Use -xls followed by a number 1 (`-xls 1`) to open Excel with the data. There is a newer -xlsx option.

-xls95

Saves the Excel file in Excel 5.0/95 format. Default format is for the version of the Excel installed on the machine building the Excel file.

-xlscolor

Uses the cell background color in Excel. Excel has a palette of 56 colors so the closest matching color is used. Avoid using a wide range of colors in your report as they may not translate well into a matching Excel color.

-xlsmail

Opens the user's email program to a composition window with the newly created Excel file attached. May not work with all email programs. Must use the -xls option to create an Excel file for this option to work.

-xlsopen

Automatically opens Excel and loads the newly created Excel file. Must use the -xls option to create an Excel file.

-xlsx *filename.xlsx*

Creates an Excel file using the filename specified. It is recommended that you fully qualify the path.

-xps *filename.xps*

Creates an XPS file using the filename specified. It is recommended that you fully qualify the path or the file will end up in your default document area (i.e. "My Documents" directory). See the XPS Document section for more information.

-xpsmail

Opens the user's email program to a composition window with the newly created XPS file attached. May not work with all email programs. Must use the -xps option to create an XPS file for this option to work.

-xpsopen

Automatically opens the default XPS viewer and loads the newly created XPS file. Must use the -xps option to create an XPS file.

Executable Examples

The executable is designed to run from a command prompt. The server version of the executable can run without any user interaction in a batch/script file or on a web site. The program is pdfrw.exe (or pdfrw64.exe) on Windows systems and pdfrw (or pdfrw64) on Linux/Unix.

Run "pdfrw sample.frw" to create sample.pdf from the sample input file. The program defaults the output file name to the same as the input name when the output name is left off.

You may also send input to the program from standard input (STDIN). Use a dash as the file name in this case. For example:
"pdfrw - myfile.pdf"
Then type in Report Writer commands and end with a single dot or ctrl-D (ctrl-Z on DOS). Using this approach you can also pipe a file into the program like this:
"cat sample.frw | pdfrw - myfile.pdf -force"

You may also call the program from a web site and stream the PDF back to your program or browser. Store your commands in the environment variable PDFCMD (case sensitive). Your system may impose limits on how much data can be stored in an environmental variable or the total for all environmental variables. You may use a new line character (ASCII 10) to break up the commands. Next, execute the pdfrw program from your application. The contents of the PDF will be printed to the standard output (STDOUT). You may capture the output or simply let it print.

Here's an example of a Perl program running on a web server streaming the PDF to the browser. The output is first stored in the $pdf variable.

```
#!/usr/bin/perl

$ENV{'PDFCMD'} = "<PDF>\n";
$ENV{'PDFCMD'} .= "<PAGE>\n";
$ENV{'PDFCMD'} .= "<TEXT ALIGN=C SIZE=20>\n";
$ENV{'PDFCMD'} .= "Report Writer Test\n";
$ENV{'PDFCMD'} .= "</TEXT>\n";

open(PDF, "./pdfrw -o passwd -noprint |");
binmode(PDF);
undef $/;
my $pdf = <PDF>;
$/ = "\n";
close(PDF);
print "Content-type: application/pdf\n\n";
print $pdf;
```

Here's another example that simply calls the program without capturing the output. The output is still streamed to the browser.

```
#!/usr/bin/perl

$ENV{'PDFCMD'} = "<PDF>\n";
$ENV{'PDFCMD'} .= "<PAGE>\n";
$ENV{'PDFCMD'} .= "<TEXT ALIGN=C SIZE=20>\n";
$ENV{'PDFCMD'} .= "Report Writer Test\n";
$ENV{'PDFCMD'} .= "</TEXT>\n";

print "Content-type: application/pdf\n\n";
exec("./pdfrw -o passwd -noprint");
```

Here's a PHP function to call Report Writer and have the results show in the browser. No output file is created in this case (since 'stdout' is used as the output file name) but you will need to create and pass in the input file (the $filein parameter).

```
<?php
function pdfrw($filein, $options = "")
{
  header("Content-Type: application/pdf");
  flush();
  passthru("/path-to-rw/pdfrw \"$filein\" stdout $options");
}
pdfrw("/path-to-input/filein.frw","-o pwd");
?>
```

Here's another PHP function to store the report commands in the PDFCMD environment variable as input. Note that -none must be the first parameter to tell the program there is no input file. Again, no output file is created.

```
<?php
function pdfrw2($options = "")
{
  header("Content-Type: application/pdf");
  flush();
  passthru("/path-to-rw/pdfrw -none stdout $options");
}
$evar = "<PDF>\n";
$evar .= "<PAGE>\n";
$evar .= "<TEXT ALIGN=C SIZE=20>\n";
$evar .= "Report Writer Test\n";
$evar .= "</TEXT>\n";
putenv("PDFCMD=$evar");
pdfrw2();
?>
```

Using the Linux Executable

The Linux version contains the same command set as the Windows version. The difference is the options that relate to the Windows API (such as auto open, print, Excel features) are not available. Instead, there are other options (-openscr, -printscr and -mailscr) that you can use to specify a script. Report Writer will pass the output PDF path and file name to the script you specify via those options. You may then perform whatever processing you want for these functions on your system.

Running the program from the command line works the same as the Windows executable. The executable program is pdfrw or the 64-bit pdfrw64. Be sure you have it marked as executable using the chmod command if necessary. For example, "chmod 775 pdfrw" from the command prompt.

Using the DLL (Dynamic Link Library)

The DLL is compiled as a 32 and 64-bit .NET DLL. You may register it with regasm to make it available from a program that uses a COM DLL such as VBScript or PHP. The file pdfrw_20.dll is the compiled DLL. The source code is available on GitHub if you want to make changes.

The .NET DLL is a wrapper for the executable pdfrw.exe or pdfrw64.exe (which is the default). It allows you to easily add the functionality of Report Writer to any existing code that can access a .NET or COM DLL. If you start a Report Writer server then the DLL may be used to call the running server instead.

The DLL also has methods to start and stop a Report Writer server though you may also do so from the command line (see the -server option). This provides the added benefit of keeping the program in memory for quick access without the cost of program startup and shutdown each time you build a PDF. In addition you can control the number of simultaneous builds to minimize memory or CPU usage. If your license allows, you may run multiple Report Servers on the same or different boxes and the DLL will cycle requests between the running servers. Each server instance for Report Writer is a separate license when purchasing individually.

Use the startServer method to bring up a Report Writer server. Report Writer then waits in memory listening for commands on a port that you specify when starting the server. The DLL sends commands to Report Writer on that port in order to build the PDF. You may have Report Writer save the PDF to disk or send it back to the DLL as byte array.

licInfo String licName ,String licPwd ,int autoDownload	Specify your license name, password and optionally if you want to download the license configuration automatically. You only need to specify this on the server startup if you are using the startServer method in which case users of the server do not need to include this information.

object buildReport bool waitForExit = true String saveFile = ""	Call this method to build the PDF after setting all of your other options. The server will be used if setServer or setServerFile was used. Set the first parameter to true when you want to wait for Report Writer to finish building the PDF before returning from this method when using the executable (non-server) version. The waitForExit is used to mean wait and return the byte array of the PDF when using the server version. Optionally pass a saveFile which will be used to save the file locally when running as a server. You may also set saveFile when not using a sever to bypass the setOutFile method call. Returns an object (when waitForExit is true). The object has two properties: bytes[] Bytes String Msg The Bytes property is the raw bytes of the output PDF. Use this to store in a database or stream to a browser. The Msg is a message with an error or "OK" if PDF was generated (though may still have issues if bad information was passed) without error.
resetOpts(bool resetServer = false)	If you using the DLL object to build more than one PDF in the same instance you should call this method to clear your current settings. That is, any DLL methods you have called such as setPDFCmd so you may set them again without retaining any settings from the previous run. Optionally pass true to reset any server information you sent with setServer.

sendFileTCP String fileName, ,String filePath = ""	The name and location of a file to send when using PDF Report Writer as a service. Use this when the server is on a different box or you don't have knowledge of what directory the service is running under. For example, you might use setInFile("somefile.frw") where you are using "somefile.frw" as a placeholder. You then call sendFileTCP "somefile.frw", "c:\temp\myrealfile.frw". This sends c:\temp\myrealfile.frw to the server and instructs it to treat it as a file named somefile.frw.
serverStatus (bool allServers = false)	Provides a status on the server (or all servers). Use setServer first to assign the host and port if they differ from the default.
setAllowBreaks	Allows use of line breaks within input tags. Line breaks are only allowed between options and not within an option (including the option word, and equal sign and value if the option takes a value). For example: <TD WIDTH=30 NOWRAP> can be written as: <TD WIDTH=30 NOWRAP>
setAllowPerl	Allows parsing of Perl code in the input (layout) file. See the XML Data Merge section for information on using the PERL tag.
setAutoSendFiles	This is optionally used when running a Report Writer server on a different box from where the request is being made. Used to send all the input files without using setSend for each one. You only need to set this option once. When the server program looks for a file and this option was used it will send a request back to the client requesting the file. The assumption on the server is none of the files being processed are local files.

setCmdlineOpts(String text)

Pass any additional command line prompts similar to how you would with the executable version of the program. For example, "-debug \"somefile.txt\" -comp15". These may contain commands that already have a corresponding method if you prefer to add them this way.

setComp15

Uses a compression algorithm compatible with PDF 1.5 (Acrobat 6.0). PDFs with this form of compression can be viewed only with Acrobat or Reader version 6 or higher. The reduction in size is based on the number and type of objects in the PDF but in general is around 10-20%. Not all PDFs will be reduced by the same percentage factor.

setDataCmd (String text)

Sets the data commands to use with the input (layout) file. Use this method if want to pass the data directly from your program rather than use a file with the setDataFile method.

setDataFile (String path-file)

Sets the data file to use with the input (layout) file. May be a file on the system or web address of a file. See the XML Data Merge section for information on merging a layout file with a data file.

setEncrypt128

Sets RC4 128-bit encryption method. Files encrypted with 128-bit encryption can only be opened with Acrobat or Adobe Reader 5.0 or above. The default encryption is 40-bit that works with Acrobat and Adobe Reader 4.0 and above.

setEncryptAES ("128" or "256")

Sets AES encryption method. Pass 128 for 128-bit encryption or 256 for 256-bit encryption. Files encrypted with AES 128-bit encryption can only be opened with Acrobat or Adobe Reader 7.0 or above. Files encrypted with AES 256-bit encryption can only be opened with Acrobat or Adobe Reader 9.0 or above.

setGUIOff

Suppresses the dialog window that shows the current build progress.

setInFile (String path-file)

Full path and name of the input file. You set the input file only if you want to read the commands from an existing file (as opposed to using setPDFCmd).
Leave this out if you are using setPDFCmd to pass in your input layout.

setKeyCode(String keyCame)

Specify your key code to use when running a licensed version. You only need to specify this on the server startup if you are using the startServer method in which case users of the server do not need to include this value.

setKeyName(String keyName)

Specify your key name to use when running a licensed version. You only need to specify this on the server startup if you are using the startServer method in which case users of the server do not need to include this value.

setNoAnnote

Disables add/change of form fields or annotations. Use with the setOwner or setUser method.

setNoChange

Disables changes to the document. Use with the setOwner or setUser method.

setNoCopy

Disables copying of text and/or graphics from the document. Use with the setOwner or setUser method.

setNoPrint

Disables printing of the document. Use with the setOwner or setUser method.

setOptimize(bool compress = true)

Optimize (linearize) the output PDF for fast web viewing. Pass true to further compress the contents for a slightly smaller PDF. Note this typically increases the size of the output by a few hundred bytes or so. The PDF is optimized for viewing on the web as opposed to shrinking the physical size. Additionally, you must create the PDF to a file rather than stream the output to the browser. The setting "fast web view" will be set to yes for optimized PDFs when you open in Reader and check the properties. This means the first page of the PDF is sent to the user and made viewable while the rest of the pages continue to download in the background.

setOutFile (String path-file)

Full path and name of the output file. Leave this method call out if you just want to receive the byte array of the PDF. This is the path and file name to save the output PDF. When using a server this is treated as the path and file name on the box where the Report Writer server is running.

setOwner (String password)

Sets the owner password for the PDF. If not specified but the user password is, this is set to the user password. Also, when not specified, the owner has only the rights granted when the document was created. So for example, if setNoPrint was specified, then it is impossible for the owner to print the document.

setQuick

Quick build - performs fewer passes through the file but some features, such as links, will not be available. See the Build Options section for more information. This option builds the PDF in memory first.

setQuick2

Quick build method 2 - performs fewer passes through the file but some features, such as total number of pages, will not be available. See the Build Options section for more information. This option builds the PDF to disk (rather than memory first) when an output file is used. It's best to use this method over setQuick unless you are using the &totpage variable.

setServer
(String host = "localhost",
int port = 7075)

Assigns the host and port for the server to use for the duration of the connection to the DLL object. The defaults are shown above. For example: setServer "192.168.0.5", 7090

setServerFile (String fileName)

A list of IP addresses and port numbers where Report Writer servers are running. The DLL will cycle between these for requests. The list of servers are stored in a static list in the DLL meaning once set all users that access the DLL will use the same settings without having to include this method call. Lines that begin with # are ignored. Use exe for the IP port to specify where the executable is located when not running a server. For example:
localhost 7090
192.168.0.5 7090
192.168.0.8 7095

Assigns three servers (note you must have a license for each server that is running Report Writer). To instead specify where the executable is located when not using a server enter a line like this:
exe c:\mypath\pdfrw64.exe

setUser (String password)

Sets the user password for the PDF. Using this method will require a password to be entered in order to open the PDF.

```
startServer (String host = "localhost",
 int port = 7075,
 int pool = 5,
 String log = "")
```

Starts a PDF Report Writer server on the box you are issuing this command from. The default values are shown above. The default executable is pdfrw64.exe. If you want to use pdfrw.exe (32-bit) or need to specify the path where pdfrw64.exe is located then use setExe(path-file). Additionally, you'll need to include a key-name/key-code combination or have a software subscription. Use setKeyName/setKeyCode or licInfo methods to specify. You may also use setKeyName("demo") if you are using the demo version of the software. The "log" option is the path and file name of the file for the server log output. If you running a server for access by other computers then do not use localhost. Instead, use the name of the box or its IP address.

stopServer

Stops the PDF Report Writer server. Use setServer first to assign the host and port if they differ from the default.

Example

Here is a small example of calling the DLL using VBS.

```
Dim PDF, outPdf
Set PDF = CreateObject("FyTek.ReportWriter")
PDF.setOutFile "c:\temp\hello.pdf"
PDF.setPDFCmd ("<PDF>")
PDF.setPDFCmd ("<PAGE>")
PDF.setPDFCmd ("<TEXT ALIGN=C>")
PDF.setPDFCmd ("Hello, world")
PDF.setPDFCmd ("</TEXT>")
PDF.buildReport()
```

Visit GitHub for more sample code.

Structure

A document may contain one or more reports. Each report is made up of a header, footer, subheader, subfooter and a body section. You can also create a report without any of the above sections but a typical report will probably have at least a header and body section. A page layout showing all of the possible sections is shown in the following table.

HEADER
SUBHEADER
BODY
SUBFOOTER
FOOTER

You may place as many reports as you wish in each document. Each report may have different layouts, different header/footer, etc. The set of commands below would consist of one report.

<HEADER>...</HEADER>
<FOOTER>...</FOOTER>
<BODY>
<SUBHEADER>..</SUBHEADER>
<SUBFOOTER>..</SUBFOOTER>
...Text and tables for report... (will autobreak across pages)
<PAGE> (only if want to force a page break and/or redefine subheader/subfooter)
<SUBHEADER>..</SUBHEADER> (only if you want to redefine the subheading at this point)
<SUBFOOTER>..</SUBFOOTER> (only if you want to redefine the subfooter at this point)
...Text and tables for report... (will autobreak across pages)
</BODY>

Commands

All commands must be enclosed in angle brackets. Commands may be entered in upper or lowercase.

Any text between tags (unless part of an opening/closing tag) is ignored.

Tags may not span lines unless you use the -allowbreaks executable option or the setAllowBreaks DLL method. You may, however, place as many tags as you wish on a single line.

Line breaks (when set to allow them) may only occur between options and not in the middle of an option or between the option and its value. For example:

```
<TD WIDTH=30 NOWRAP BGCOLOR="red">
```

is valid whether or not line breaks are set as allowed. When allowing line breaks, the above can be written as:

```
<TD
 WIDTH=30 NOWRAP
 BGCOLOR="red">
```

or any combination where the line breaks occur between the options.

This, however, is not valid since the WIDTH and its value are split and the BGCOLOR option is split:

```
<TD
 WIDTH=
 30 NOWRAP
 BG
 COLOR="red">
```

Use single or double quotes when entering a text value with spaces as a parameter. For example, <TAG DESCR="My Test Description">.

Do not leave a space between a parameter name, the = sign and its value.

Correct	<PAGE WIDTH=8.5 HEIGHT=11>
Incorrect	<PAGE WIDTH = 8.5 HEIGHT= 11>

Any text not inside of a TEXT block or TABLE is treated as a comment. You may also place comments inside of TEXT blocks but they must be enclosed in a tag which starts with <! and ends with >.

Use a slash in front of the " or ' character in the situation where it's part of the string. For example, to use the string 'Mike's "Slightly Used" Cars' as a string parameter to an input statement, write it using one of the following methods:

```
<INPUT NAME="compname" TYPE="text" VALUE="Mike's \"Slightly Used\" Cars">
<INPUT NAME="compname" TYPE="text" VALUE='Mike\'s "Slightly Used" Cars'>
```

Note that either the " or ' character can be used to enclose a string. You must use the corresponding character to close the string that you used to open it with however.

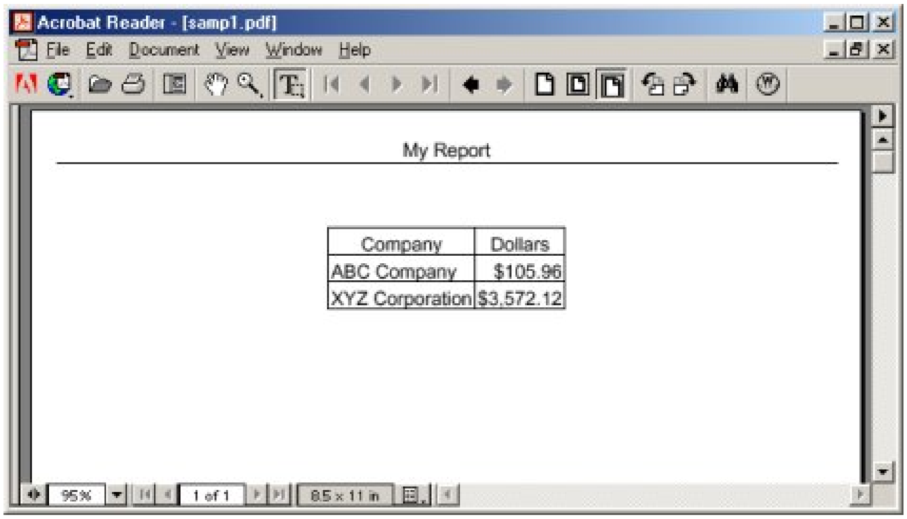
You may also use quotes around numeric values and use a /> to close a tag. This is for compatibility with an XML syntax approach. Any of the following are acceptable tags for Report Writer and all work the same:

```
<page height=11 width=8.5>
<page height="11" width="8.5"/>
<page height="11" width="8.5" />
```

Here is a simple report to give you an idea of how the commands look:

```
<PDF>
<PAGE>
<GRID X=.5 Y=.5 WIDTH=7.5 HEIGHT=10 XUNITS=80 YUNITS=80>
<FONT FACE=2 SIZE=11>
<HEADER MARGIN=5>
<TEXT ALIGN=C SIZE=14>
My Report
</TEXT>
<LINE X1=0 X2=80>
</HEADER>
<BODY>
<TABLE BORDER=1 TBLALIGN=C>
<TH ALIGN=C><TD>Company</TD><TD>Dollars</TD></TH>
<TR><TD>ABC Company</TD><TD ALIGN=R>$105.96</TD></TR>
<TR><TD>XYZ Corporation</TD><TD ALIGN=R>$3,572.12</TD></TR>
</TABLE>
</BODY>
```

Which creates the following output:


Acrobat Reader - [samp1.pdf]
File Edit Document View Window Help
95%
1 of 1
8.5 x 11 in

My Report

Company	Dollars
ABC Company	$105.96
XYZ Corporation	$3,572.12

Document Level Commands

Document level commands are used to apply certain settings or properties to the document. The title of the document or the zoom factor to use when opened, for example. All the commands in this section are optional except for the opening <PDF> tag. This must be the first command for Report Writer to treat the input as a set of Report Writer mark-up commands.

```
<ADDFONT
    NAME=text
    SRC=text
    ID=text
    AUTOBI
    ENCODING=text
    CODEPAGE=text
    BOM=text
    NOEMBED
    UNICODE
    SUBSET
    LINEBREAK=text
    CHRWIDTH=number
    DEFAULT>
```

Add font command - adds (embeds by default) a TrueType, OpenType or Type 1 font in the document. You may specify just the name attribute on Windows based systems and the software will locate the font from the registry. For example, `<ADDFONT NAME="verdana">`. For other operating systems (optionally on Windows) you must use the SRC option to specify the location of the font file on disk.

Parameter	Description
NAME=text	The name of the font (such as "Times New Roman"). This is also the name displayed in Adobe Reader when Document Info|Fonts is selected from the menu. Specify the name of the font (such as Verdana or Tahoma) on Windows based systems and Report Writer will locate the font in the registry so you don't have to specify the SRC value.
SRC=text	Optional on Windows based systems if the NAME attribute specifies an installed font. Required on all other platforms. For TrueType fonts, the font file (ex. "c:\windows\fonts\myfont.ttf"). Also, for OpenType fonts, the font file (ex. "c:\windows\fonts\myfont.otf"). You may also specify a file from the web in the form http://www.mysite.com/myfont.ttf. For Type 1 fonts, the file name without the extension (ex. "c:\windows\fonts\myfont"). Type 1 fonts have several different files associated with them and the software will handle locating the individual files.

Parameter	Description
ID=text	Enter a name for the font to refer to it by in your report. The default for this option, if it's not specified, is the NAME option. You can use spaces in the name, just be sure to put quotes around it. This is the value you would use for the FACE parameter in the FONT command.
AUTOBI	Auto Bold-Italics option. Use this option to load only one version of the font and have PDF Report Writer create any needed bold or italics text when the <B> or <I> tags are used.
ENCODING=text	The encoding to use. WinAnsiEncoding is used if not specified. This value is inserted directly into the PDF as typed so case is important. If you are not sure what value to use, leave this option out. The default should be fine for most cases. Possible values are WinAnsiEncoding, StandardEncoding, MacRomanEncoding or PDFDocEncoding.

Parameter	Description
CODEPAGE=text	The codepage to use (1252 Windows Latin-1 is used by default). This option is valid only when adding your own TrueType or OpenType font. Must be a codepage that is included in the font. Currently, the other codepages supported by Report Writer are: IDENTITY (None - text is Unicode)[1,2] 737 (Greek)[1] 855 (Cyrillic)[1] 857 (Turkish)[1] 862 (Hebrew)[1] 866 (Cyrillic)[1] 869 (Greek)[1] 874 (Thai)[1] 932 (Japanese)[1] 936 (GBK - Chinese)[1] 949 (Korean)[1] 1250 (Central European) 1251 (Cyrillic) 1253 (Greek) 1254 (Turkish) 1255 (Hebrew) 1256 (Arabic) UTF8 (UTF-8)[1] [1] Use the UNICODE option with these code pages and, typically, SUBSET [2] All characters using this font, even those which can be represented by a single byte, must be double byte. See the META tag to specify UTF-8 encoding for your text. The codepage does not need to be specified for UTF-8 text. The font must contain the desired characters, however.
BOM=text	The Byte Order Mark for use with Unicode text and IDENTITY as the CODEPAGE value. Set this to the text "FFFE" if the Unicode text is in little-endian order. For example, if your text is the two-byte hex string D5 5C for the Korean symbol 한 then you do not need to use this option. If your string is 5C D5 then set BOM="FFFE".

Document Level Commands

Parameter	Description
NOEMBED	Specifies the font is not to be embedded into the PDF. The resulting PDF will be smaller but the end user of the PDF must have the font installed on their system in order to properly view the PDF. The NAME option is used to determine the corresponding system font. NAME is case sensitive when using NOEMBED (unless you're using Windows and the font information is being read from the registry). This option is only recommended when you need to save space and you have some control over the target audience (such as for your own personal use or in-house use). As an alternative, see the SUBSET option to include only those glyphs from the font used in your PDF.
UNICODE	Allows you to specify Unicode text with the font. Unicode text must be formatted as UTF-8 or use the syntax &#9999; where 9999 is the decimal Unicode value, &#x9999; where 9999 is the hexadecimal Unicode value or &#o9999; where 9999 is the octal Unicode value. For example, &#1575; is the same as &#x0627; and &#o3047;. The font file must include the Unicode characters in order to display them (such as the arialuni.ttf font supplied with Windows). For example, the codes &#x05E9; &#1610; &#x2190; &#x2191; &#x2192; &#x2193; &#x2194; &#x2195; will be displayed as: ש ي ← ↑ → ↓ ↔ ↕
SUBSET	Use this option to subset the font. This will reduce the PDF size by including only those glyphs from the font that are used in the PDF.
LINEBREAK=text	For use with UNICODE text. Enter a string of 2-bytes (or a series of 2-byte strings) to use as a line break character. You may specify multiple glyphs by placing one after another so that the entire length of this entry is an even number. The text entries should be based on the CODEPAGE. For example, you might set this to the hex values A1 A3 when using CODEPAGE 936.

Parameter	Description
CHRWIDTH=number	Use this to force the width of all characters to be the same, treating a variable width font as fixed-width. Since the sizes are being set differently than the default the letter spacing will likely not look good. The purpose is more for aligning numbers where your input may have been formatted for a fixed-width font. The width is measured in units in which 1000 units corresponds to 1 unit in text space. Numbers in the range of 500 to 600 should be good but you may need to vary based on your font.
DEFAULT	Specifies this font is the default to use. Only one ADDFONT tag should have this option set. The DEFAULT used will be the last font added with this option if multiples are found.

PDF Report writer has 14 built-in fonts. When you add fonts, they are numbered starting at 15. The first ADDFONT command sets that font to number 15. The next ADDFONT command sets that font to number 16 and so on. You then use these numbers, or the ID value described above, as the FACE parameter for the FONT tag.

Example:

```
<PDF>
<ADDFONT SRC="c:\winnt\fonts\myfont.ttf" ID="my font">
<ADDFONT SRC="c:\winnt\fonts\myfont">
<ADDFONT NAME="tahoma">
<ADDFONT NAME="Lucida Handwriting" NOEMBED>
<PAGE>
<TEXT FACE=15 SIZE=12>
Here is the TrueType font embedded in the document.
<FONT FACE=16>
Here is the type 1 font embedded in the document.
<FONT FACE="my font">
Here is the TrueType font again using the ID instead of the number.
<FONT FACE="tahoma">
This TrueType font was added by locating its source in the Windows
registry.
<FONT FACE="Lucida Handwriting">
This TrueType font was not embedded - end users must have this font
installed.
</TEXT>
```

<AUTHOR
LANGUAGE=text
COUNTRY=text>
</AUTHOR>

Sets the text for author of the document. Place the text between the opening and closing tags.

Parameter	Description
LANGUAGE=text	Text is assumed to be Unicode (2-character format) when this is used. A 2-character ISO 639 language code - for example, EN for English or JA for Japanese. The complete list of codes are available through http://www.iso.ch.
COUNTRY=text	Optional, used with the LANGUAGE option above. A 2-character ISO 3166 country code - for example, US for the United States or JP for Japan.

**<BASEALIGN
VALUE=L|R|C|J>**

Sets the base alignment for all table cells in the document.

Parameter	Description
VALUE=L|R|C|J	Sets the default alignment to Left, Right, Center or Justify. Table cells default to Left alignment without this command.

**<BREAKON
VALUE="text">**

Used to specify a list of characters in addition to the space character to line break on. For example, set VALUE="-" to include the dash as a valid character to break on. You may string together any combination of characters but don't place any separator in the list (comma or space).

Parameter	Description
VALUE="text"	The set of characters line breaks may occur on in addition to the space character.

<CACHEALLIMG>

Sets the CACHE option for all images in the PDF. See the IMG tag for details on the CACHE option.

<CLOCK
VALUE=number>

Used to set the clock to a 24-hour (military) format (see the time variable) Default is a 12-hour clock using am/pm.

Parameter	Description
VALUE=number	Set to 24 for a 24-hour clock format. Any other value sets a 12-hour clock format.

<COLOR
NAME=text
VALUE=color>

Used to name a color. You can create a name for a color then refer to that color by name throughout your document. You just change the color value in one place if you decide to change color schemes later.

Parameter	Description
NAME=text	The name for the color. Place the name in quotes if you use spaces in the name.
VALUE=color	Any valid color code.

```
<CREATOR
    LANGUAGE=text
    COUNTRY=text>
</CREATOR>
```

Sets the text for creator of the document. Place the text between the opening and closing tags.

Parameter	Description
LANGUAGE=text	Text is assumed to be Unicode (2-character format) when this is used. A 2-character ISO 639 language code - for example, EN for English or JA for Japanese. The complete list of codes are available through http://www.iso.ch.
COUNTRY=text	Optional, used with the LANGUAGE option above. A 2-character ISO 3166 country code - for example, US for the United States or JP for Japan.

<DATASET
ID=text
SKIPIFEMPTY>
</DATASET>

Sets the area in the layout where data will be placed. The DATASET block should surround a row or rows when using tables if there will be multiple entries in the data.

Also used in the data file to set a data block. See the XML Data Merge section for more information.

Parameter	Description
ID=text	The ID of the data block.
SKIPIFEMPTY	Do not render the section in the output if the section is empty in the XML.

<FIRSTOF
    ID=text
>
</FIRSTOF>

For use with the DATASET tag. This block goes in between the opening and closing DATASET tags to only include the section if this is the first iteration of the dataset block. This is used to setup a table and header for an inner DATASET so it is only rendered on the first record and not for every record. See the XML Data Merge section for more information.

Parameter	Description
ID=text	The ID of the data block.

```
<LASTOF
    ID=text
>
</LASTOF>
```

For use with the DATASET tag. This block goes in between the opening and closing DATASET tags to only include the section if this is the last iteration of the dataset block. Use this to setup some footer text for an inner DATASET so it is only rendered on the last record and not for every record. See the XML Data Merge section for more information.

Parameter	Description
ID=text	The ID of the data block.

<IFBETWEEN
ID=text
FROM=number
THRU=number>
</FIRSTOF>

For use with the DATASET tag. This block goes in between the opening and closing DATASET tags to only include the section if the current iteration of the dataset is between the FROM and THRU value specified. The staring interation is one. Use negative numbers to mean from the end of the array of dataset values. For example, FROM=-2 THRU=-1 means the last 2 entries. See the XML Data Merge section for more information.

Parameter	Description
FROM=number	The starting record count in the array of dataset values where 1 is the first entry. Use a negative number to specify from the end of the array set where -1 is the last entry in the dataset.
THRUE=number	The ending record count in the array of dataset values where 1 is the first entry. Use a negative number to specify from the end of the array set where -1 is the last entry in the dataset.
ID=text	The ID of the data block.

<DELETEPDF
PAUSE=seconds
TIMEOUT=seconds>

Deletes the output PDF once the program is finished. This is useful when emailing the PDF and you don't want to retain a copy of the PDF that was sent. Do not use in situations where you need the PDF to remain available - such as when you're automatically opening the PDF. When emailing, the PDF will be sent or attached first so you can use this tag in those situations.

You can optionally supply the number of seconds to wait before deleting the PDF. Use this in cases where you want to print it and need to wait a few seconds for that to happen. Also, you can set a timeout value to specify how long to keep trying to delete the PDF. These options (when used) can be used individually or together. For example, setting PAUSE=10 and TIMEOUT=60 means to wait 10 seconds then try to delete the PDF. If not successful in deleting right away, keep trying for up to 60 seconds.

Parameter	Description
PAUSE=seconds	The program will pause for the specified number of seconds then attempt to delete the PDF. The number of seconds must be an integer.
TIMEOUT=seconds	Sets the maximum amount of time to use in attempting to delete the PDF. The number of seconds must be an integer.

<DIV
DIR=text>
</DIV>

Used to denote the reading direction. Setting to RTL will reverse the column ordering of tables. In addition, any UTF-8 characters will be reversed in their individual words and sentences. This option is useful for Hebrew and Arabic text.

Place any TEXT or TABLE tags between the opening and closing DIV tags. Be sure to include the closing </DIV> tag.

Parameter	Description
DIR=text	The reading direction. Set to RTL for Right to Left. Using this option may result in significantly more processing time depending on how much text is in your document. Any other value assumes the default left to right direction for text.

<EMBED
SRC=text
MIME=text
FILENAME=text
SUBJECT=text

Embeds an external file into the PDF at the document level. Similar to the EMBED option with the A tag except there is no visual indicator on any page for this embedded document.

Parameter	Description
SRC=text	Required. The path and name of the file to embed. You may also specify a file from the web in the form http://www.mysite.com/include.frw.
MIME=text	The mime type for the file. This is determined for some files automatically by the file name extension. The files currently configured are .avi, .wmv, .mpeg, .mov, .swf, .wav, and .mp3. Other types must supply the value (unless it exists in the Windows system registry at build time). For example, video/x-msvideo or audio/x-midi. Windows based versions of Report Writer will attempt to locate the correct mime type from the system registry when the PDF is built.
FILENAME=text	The file name to use for the embedded file. This is not the source file name on disk - it is the filename to refer to the embedded file by once it's in the PDF. The default is the name used in the SRC entry.
SUBJECT=text	The subject the end user will see for the embedded file.

<ENCRYPT
OWNER=text
USER=text
NOPRINT
NOCHANGE
NOCOPY
NOANNOTE
128
AES=number>

Used to place encryption/password protection on the PDF. Command line options for encryption override this tag.

Parameter	Description
OWNER=text	Sets the owner password for the PDF. If not specified but the user password is, this is set to the user password. Also, when not specified, the owner has only the rights granted when the document was created. So for example, if NOPRINT is specified, then it is impossible for the owner to print the document.
USER=text	Sets the user password for the PDF. The following four options can be used to turn off various features for the user.
NOPRINT	Disables printing of the document.
NOCHANGE	Disables changes to the document.
NOCOPY	Disables copying of text and/or graphics from the document.
NOANNOTE	Disables add/change of form fields or annotations.
128	Use RC4 128-bit encryption. Files encrypted with 128-bit encryption can only be opened with Acrobat or Adobe Reader 5.0 or above. The default encryption is 40-bit which works with Acrobat and Adobe Reader 4.0 and above.
AES=number	Use AES=128 for AES 128-bit encryption or AES=256 for AES 256-bit encryption. AES 128-bit encrypted PDFs can only be opened with Acrobat or Adobe Reader 7.0 and above. AES 256-bit encrypted PDFs can only be opened with Acrobat or Adobe Reader 9.0 and above.

<EXCEL
VALUE=text
COLOR>

Specifies that all text and tables should be stored for export to Microsoft Excel or CSV. Using this option or the EXCEL option on individual text blocks or tables will store the text or cell contents in a separate area of the PDF for retrieval by FyTek's PDF to Excel coverter. Leave this tag out and use the EXCEL option on individual text blocks or tables if you only want certain sections exported.

You can specify an Excel output file and/or a CSV file to create along with the PDF file. This type of file does not require Excel in order to create the file. For example, the following command (using the exe version) will create both a PDF (called fileout.pdf) and a CSV file (called fileout.csv):

```
pdfrw.exe filein.frw fileout.pdf -csv fileout.csv
```

Use the -xls option on the exe version of Report Writer to create an Excel file. For example, the following command (using the exe version) will create both a PDF (called fileout.pdf) and an Excel file (called fileout.xls):

```
pdfrw.exe filein.frw fileout.pdf -xls fileout.xls
```

Use the -html option on the exe version of Report Writer to create an HTML file. This option is only for data export and does not create an HTML file formatted to match the PDF. Note that with newer versions of Excel you may open HTML files directly. Use this as an alternative to the -xls command which uses OLE and require Excel be loaded on the machine running Report Writer. Name the output with a .xls extension in this case rather than .html. For example, the following command (using the exe version) will create both a PDF (called fileout.pdf) and an HTML file (called fileout.html):

```
pdfrw.exe filein.frw fileout.pdf -html fileout.html
```

Be sure to fully qualify the path you want for the Excel file or it may wind up in the "My Documents" area. You can have Excel open with the data automatically by passing in the number 1 instead of a file name on the -xls option. You can always pull the Excel/CSV data later from the PDF using the PDF to Excel converter program available on FyTek's web site. The converter is the program pdf2excl.exe and may be distributed freely. Look for the download on the same HTML page where PDF Report Writer is located.

Parameter	Description
VALUE=text	Set to NOSTORE to not store the extra data required to extract to Excel or CSV in the PDF file for later retrieval. This allows you to create an Excel, CSV or HTML file while you are creating the PDF but the extra data won't be there to extract later. This makes for a slightly smaller PDF.
COLOR	Uses the cell background color in Excel. Excel has a palette of 56 colors so the closest matching color is used. Avoid using a wide range of colors in your report as they may not translate well into a matching Excel color.

<FDFDATA
NAME=text
FACE=font
SIZE=number
COLOR=color
BGCOLOR=color
BORDERCOLOR=color
BORDERSTYLE=text
BORDERDASH=text
BORDERWIDTH=number
XOFF=number
YOFF=number
VALUE=text
HIDDEN
NOPRINT
NOVIEW
FDFFIXED>

-- or --

<FDFDATA
...Any options above except VALUE...>
Value Text
</FDFDATA>

Used to fill in a form field from the included PDF. The included PDF may be created from Report Writer or another program, such as Acrobat. Fields should be created in the original PDF as readonly if you don't want the end user to modify values. You may either specify the value using the VALUE option on the FDFDATA tag or place a multi-line value between the <FDFDATA> and </FDFDATA> tags. Set VALUE equal to any value to turn on a checkbox. Text will break where you specify the breaks in mutli-line fields. There is no auto-wrapping of text.

May want to use the VIEWERAPP option with the FDFFIXED tag to have the viewer determine the best display for the data.

Only text, textarea and checkbox widget types are currently supported for this feature. See the backgrounds section for more information. Also see the PDFPAGE command.

Parameter	Description
NAME=text	The name of the field. The name is case sensitive.
FACE=font	Optional. The font to use. Only when using FDFFIXED as well.

Document Level Commands

Parameter	Description
SIZE=number	Optional. The font size to use.
COLOR=color	The text color to use for the field.
BGCOLOR=color	The background color to use for the field. Only works when VIEWERAPP is not specified.
BORDERCOLOR="text"	The border color.
BORDERSTYLE="text"	Type of border to draw. Options are: S - Solid (Default) D - Dashed B - Beveled I - Inset U - Underline
BORDERDASH= "number number"	Two numbers separated by a space. This is used when the BORDERSTYLE is set to D (Dashed). The first number is the number of points to draw on and the second is the number of points for the gap. Default is "3 3" when BORDERSTYLE is D and this option is not specified.
BORDERWIDTH=number	Width of the border in points (1 point = 1/72 of an inch). Default is 1.
XOFF=number	An offset amount in points (1/72 of an inch) for X-axis text placement for fields when FDFFIXED is used. Default is 0.
YOFF=number	An offset amount in points (1/72 of an inch) for Y-axis text placement for fields when FDFFIXED is used. Default is 0.
VALUE=text	The value for the field. This value will overwrite any value currently stored in the field.
HIDDEN	Field is not displayed but the value is sent when the submit button is pressed.
NOPRINT	Field is not printed.
NOVIEW	Field is not displayed but is printed if the NOPRINT option is not used.
FDFFIXED	Field is removed and replaced with static text.

<FDFJSON>
JSON String
</FDFJSON>

Shortcut method using JSON to assign FDFDATA field and value pairs. Use a single set of parameters inside curly braces where the parameter name is the field name and set the value to the field value to show in the PDF. See the backgrounds section for more information. Also see the PDFPAGE command.

For example:

```
<FDFJSON>
{
  "field1[0]": "Sample text to show",
  "field2[0]": "Another text line",
  "checkboxFld1": "On",
  "cityName": "Test city"
}
</FDFJSON>
```

<FDFFIXED
VIEWERAPP=number
XOFF=number
YOFF=number
ALL>

Specifies that form text fields in FDFDATA tags should be removed and the text placed on the form. Normally, the text specified in the FDFDATA field is used to fill in the widget as the default data. The user of the PDF may then type over the data as long as the field isn't locked down. This option removes the fill in from the form altogether as if it were never there and simply treats it as text.

Parameter	Description
VIEWERAPP=number	Used to denote the viewer application sets the appearance of the fields. This is the recommended option to use with FDFFIXED. The form fields remain in the document with this option. Report Writer passes the values specified for display to the fields rather than convert the fields into text. Set to 1 to keep the fields editable or 2 to lock down the fields.
XOFF=number	An offset amount in points (1/72 of an inch) for X-axis text placement for fields not listed via FDFDATA when the ALL option is used. Default is 0.
YOFF=number	An offset amount in points (1/72 of an inch) for Y-axis text placement for fields not listed via FDFDATA when the ALL option is used. Default is 0.
ALL	Include this parameter to remove all fields - even ones not set from FDFDATA.

<FDFAPPEARANCE
FORCE=ON|OFF>

Specifies whether or not the viewer should construct the appearance for the fillable fields. You may want to use this when you pass data to fillable fields and the data is not displayed properly or lined up how you expect. This option will only work on PDFs opened with Adobe Reader or Acrobat.

Parameter	Description
FORCE=ON|OFF	Set to ON to have the viewer construct the field appearance. Set of OFF to not have the viewer construct the field appearance. The default is ON when existing fields are found in the PDF and you are not locking down (flattening) fields.

<FROMPG
NAME=text>

Used with the button input statement to specify the start page for printing. This tag is used in conjunction with the THRUPG tag to mark a page range for printing.

You may also use this tag to mark a range of pages for retrieval with the GetPageRange DLL method or for use with the OUTLINE tag.

Parameter	Description
NAME=text	The name of this page range. Place this tag where you want to mark the point to start the range. There should exist a corresponding THRUPG tag somewhere in the document with the same value. Or, you may use a closing </FROMPG> tag instead if there is no overlap of page ranges. Name values are case sensitive.

**<HTML
LANG=text>**

Optional. Sets the base language for Japanese, Chinese, or Korean when providing UTF-8 encoded text. Be sure to place this tag after the opening PDF tag.

Parameter	Description
LANG	Enter 'ja' for Japanese, 'zh' for Chinese, or 'ko' for Korean. UTF-8 characters will be converted based on the language entry. When used with the META tag, it is not necessary to specify text with one of the Japanese, Chinese, or Korean font faces. That is, you do not need to use a font face such as J1, C1, or K1. This tag is optional though it is recommended when using UTF-8 with one of these languages.

<IGNOREUNKNOWN>

Instructs the program to ignore any unknown tags. Place this tag anywhere in your input file. This is for compatibility when you have other HTML tags mixed in with your input that are not Report Writer tags.

<INCLUDE
SRC=text>

Used to include commands from an external file. The file should contain Report Writer commands and/or text. The contents of the file will be placed in the postion this command appears in the main input file. You may use as many INCLUDE commands as you wish in your input file.

Parameter	Description
SRC=text	The path and name of the input file. You may also specify a file from the web in the form http://www.mysite.com/include.frw. The file will be downloaded locally then included.

<INITIALIZE>
</INITIALIZE>

-- or --

<INITIALIZE
LINES=number>

Sets the initialization section. This tag is used to increase performance by marking a section at the top of the report as the only area which should be examined during the initialization process. Report Writer normally scans the entire input file looking for any document level options such as ADDFONT or ZOOM tags. By placing this information within these tags near the top of the report, only this subset of the input need be searched for this information. Also see the TABLESIZE option.

Place all document level commands in this section. These are the commands in this documentation under the "Document Level" bookmark.

There are two ways to use this tag. The first is to place the initialization information between opening and closing <INITIALIZE> and </INITIALIZE> tags. This allows you to place document level commands such as added fonts in this section. All information placed here, such as text, tables and images, are ignored when the final PDF output is built. This section is only used for document settings and information on fonts used outside the initialization section.

An alternate method is to use the INITIALIZE tag with the LINES option instead. There is no closing tag in this instance and, if a closing tag is found, it overrides the LINES option. The LINES option tells the program to look at the next X number of lines in the input file or set of commands. Only the following X number of lines are examined on the initial pass so all information on added fonts and document settings must be located here. Also, unlike the above method, all information such as text or images are rendered as part of the PDF just as if the INITIALIZE tag wasn't there. This feature works best when you have all document and embedded font information already located near the top of the report.

See the Build Options section for more information.

<KEEPCOLSIZE>

Automatically applies the KEEPCOLSIZE option to all TR tags that have a SUBHEADING. Only useful when using the MODTABLE tag and TR subheadings that repeat across MODTABLE tags. See the TABLE tag for more details.

<KEYWORDS
LANGUAGE=text
COUNTRY=text>
</KEYWORDS>

Sets the text for the document keywords. Place the text between the opening and closing tags.

Parameter	Description
LANGUAGE=text	Text is assumed to be Unicode (2-character format) when this is used. A 2-character ISO 639 language code - for example, EN for English or JA for Japanese. The complete list of codes are available through http://www.iso.ch.
COUNTRY=text	Optional, used with the LANGUAGE option above. A 2-character ISO 3166 country code - for example, US for the United States or JP for Japan.

```
<LAYER
    NAME=text
    OFF>
</LAYER>
```

Used to group objects (TEXT blocks, TABLES and images outside of text/table blocks) into layers. Layers are used in Acrobat and Adobe Reader 6.0 and higher. Users may selectively turn layers on or off via the layers tab in the viewer. Layers may be nested but may not cross page boundaries. Close and reopen the layer(s) if you require them to cross a page boundary.

Place your content between the opening and closing LAYER tags. Content can include a block of text (text between <TEXT> and </TEXT>), entire tables or table cells/rows. The LAYER tag may not be placed within text (inside either a TEXT block or TD table cell). The name parameter is what the user will see the layer called in the viewer. They are listed in the order they were created.

Layers can be useful in a variety of cases. For example, when you want to create one PDF file with several different logos. Place each logo in the same area on the PDF and turn on the particular one you want for printing. Another use is selectively hiding certain information, such as pricing or other financial data for printing. You may also use layers to display text in more than one language for the same PDF.

Also see the LAYERGROUP tag. Additional settings for layers are available in table rows and cells. Table cells (the TD tag) has options for setting layers and providing alternate views of the same cell depending on the active layer. See the TR and TD tags.

Parameter	Description
NAME	The case sensitive name for this layer that will appear in the viewer. This is the name the end user of the document will see. You may reuse the name on layer tags throughout your document. All areas with the same name are opened or closed together.
OFF	The default for a layer is ON, meaning the layer is visible. Setting this option causes the layer to be initially hidden when the document is first opened.

<LAYERGROUP
VALUE=text>
NAME=text>

Used to group layers together so they are treated as a radio set (only one item from the group allowed on at a time).

See the LAYER tag for information about layers.

Parameter	Description
VALUE=text	A unique identifier for the group. All of the NAME values for the various LAYERGROUP tags will be grouped together by this value.
NAME=text	The name of an existing LAYER in the document.

Here's an example grouping a set of 3 layers so only one is visible.

```
<LAYERGROUP VALUE=1 NAME=English>
<LAYERGROUP VALUE=1 NAME=Spanish>
<LAYERGROUP VALUE=1 NAME=French>
```

<LINK
REL=STYLESHEET
HREF="text">

Used to supply an external style sheet for the document. The style sheet syntax is similar to cascading style sheets used in HTML. They provide a method of visually changing the appearance of your document without changing the base document. You may use multiple LINK statements in your document.

The file may be setup with any Report Writer tag (such as TEXT, TABLE, TD) along with a group (or declaration block) of properties to apply. These properties are named the same as the options available for each tag. For example, using TABLE, you can supply items such as BORDER, FCOLOR, FACE, SIZE, etc for the options.

You may also give the group an arbitrary name and use the option CLASS with any Report Writer tag to pull in those properties. The group names (called selectors) are case insensitive.

Parameter	Description
REL=STYLESHEET	REL must be supplied and set to the string STYLESHEET.
HREF="text"	The path and file name of the style sheet. Place quotes around this value. You may also use a web address that starts with http:// to pull a style sheet from a web site.

The style sheet file contains the style name (either a single name or a comma separated list) followed by the properties enclosed in braces. The properties are named the same as the tag options. A semicolon separates each of the options and a colon is used between the property and its value. For example, assume the following is in a file called mystyle.css:

```
/*
  Style Sheet Example
*/
table,text {face:3; size:10; fcolor:red; border:1}
small {size:8}
large {size:15}
```

In your input you'd include a LINK tag with a reference to this file. Something like: <LINK REL=stylesheet HREF="c:\css\mystyle.css">

All table and text blocks will use font 3 (Times Roman), have a font size of 10, text color red and a border size of 1. Nothing special needs to be added to the TABLE or TEXT tags in the document. They will simply inherit these settings since the property group is named the same as the tag.

The groups "small" and "large" will need to be set for tags you want them used in by specifying the CLASS option. For example, to use the style "small" in a block of text you'd write <TEXT CLASS="small">. All of the properties for the default text settings (like Times Roman and color red) will be applied first. Then the specific settings for small (font size of 8) will be applied so the final text will be Time Roman, red, point size 8.

<META
CONTENT=text
RTFVIEW=text
RTFGRIDMARGINS
NO_STARTSPACE
NO_ORPHAN
NO_WIDOW>

Used to denote document level settings such as UTF-8 encoding.

Parameter	Description
CONTENT=text	The phrase "charset=utf-8" must appear somewhere in the string to specify the document contains text that is UTF-8 encoded. Use this for UTF-8 encoded input for Japanese, Chinese, or Korean.
RTFVIEW=text	The type of view or zoom level for RTF documents. The default is "4" for RTF when this option is not specified. Valid values are: 0 None 1 Page Layout view 2 Outline view 3 Master Document view 4 Normal view 5 Online Layout view
RTFGRIDMARGINS	Uses the margin settings from the GRID tag in RTF output. By default, the page margins are set to a half inch on top and a quarter inch on left, right and bottom.
NO_STARTSPACE	Removes any spaces, <BR> or <P> tags from the start of table cells. Only those found at the start of the text string for the cell (before any characters or graphics) are removed.
NO_ORPHAN	Used to apply the NO_ORPHAN setting to all TEXT blocks in the document.
NO_WIDOW	Used to apply the NO_WIDOW setting to all TEXT blocks in the document.

<METRIC
UNITS=text
VALUE=number>

Used to enter values in centimeters rather than inches. This affects the WIDTH and HEIGHT parameters of the PAGE command as well as the margin setting parameters of the GRID command.

Parameter	Description
UNITS=text	Optional. The default is centimeters when UNITS is not specified. Set to "mm" to use millimeters or "pt" for points (1 point = 1/72 of an inch).
VALUE=number	Optional. Set to your measurement unit when using something other than the above. The number entered is specified as units per inch. For example, to use decimeters you would set this value to 0.254. This value overrides any setting from the UNITS option.

**<MONTHS
VALUE=text>**

Used to set the values you want to use for names of the months (see the date variable) Default is January, February, ..., December. For example, to use German you might set this to:

```
<MONTHS VALUE="Januar,Februar,MÃ¤rz,April,Mai,Juni,Juli,August,September,Oktober,November,Dezember">
```

Parameter	Description
VALUE=text	A comma separated list of months to use.

<NOPERL>
</NOPERL>

Specifies the default Report Writer tag(s) when not using the -allowperl option or setAllowPerl method. These tags and the contents between go inside of the <PERL> </PERL> block when using XML. See the XML Data Merge section for more information.

These tags can also be used in general to specify other messages you want to print when not using the -allowperl option or setAllowPerl method. Use when you have RWIF, RWSET, or RWGET commands. See the SQL Queries section for more information on these commands as well.

<NOTEXTSCALE>
</NOPERL>

Text in tables may sometimes need to be compressed in order to fit the cell width. This option globally prevents images from being scaled along with the text.

<ABORT>

Specifies the processing of commands should stop at this point. Nothing beyond this tag will be considered part of the report. Typically, you would use this within a NOPERL block to note that the report cannot be generated because it requires Perl processing.

<OPTIMIZE>

This is the same as using the -opt option or setOptimize method.

Optimize (linearize) the output PDF for fast web viewing. Note this typically increases the size of the output by a few hundred bytes or so. The PDF is optimized for viewing on the web as opposed to shrinking the physical size. Additionally, you must create the PDF to a file rather than stream the output to the browser. The setting "fast web view" will be set to yes for optimized PDFs when you open in Reader and check the properties. This means the first page of the PDF is sent to the user and made viewable while the rest of the pages continue to download in the background.

<OPTIMIZE15>

This is the same as using the -opt15 option or setOptimize method.

Optimize (linearize) the output PDF for fast web viewing and further compresses the PDF. Additionally, you must create the PDF to a file rather than stream the output to the browser. The setting "fast web view" will be set to yes for optimized PDFs when you open in Reader and check the properties. This means the first page of the PDF is sent to the user and made viewable while the rest of the pages continue to download in the background.

```
<OUTLINE
    LEVEL=number
    DESCR=text
    CLOSED
    COLOR=color
    ITALICS
    BOLD
    LANGUAGE=text
    COUNTRY=text
    DECODE=text
    URL=text
    PRINT[=text]
    RANGE=name
    SUB_DESCR=text
    SUB_CLOSED
    SUB_COLOR=color
    SUB_ITALICS
    SUB_BOLD
    SUB_URL=text
    SUB_PRINT[=text]
    SUB_RANGE=name>
```

Used to set up an outline (bookmarks) for the document. Can also be used as a link to a web URL or to print the document.

Parameter	Description
LEVEL=number	The level of the outline. 1 is the top level, 2 would be a sub-level to 1, etc.
DESCR=text	The description that appears in the bookmarks pane.
CLOSED	Adding this option will cause the initial display of the bookmark to be closed.
COLOR=color	The color to use for the outline entry (must be an RGB color). *(Acrobat 5.0 or above)*
ITALICS	Italics outline entry *(Acrobat 5.0 or above)*
BOLD	Boldface outline entry *(Acrobat 5.0 or above)*
LANGUAGE=text	Text is assumed to be Unicode (2-character format) when this is used. A 2-character ISO 639 language code - for example, EN for English or JA for Japanese. The complete list of codes are available through http://www.iso.ch.

Document Level Commands

Parameter	Description
COUNTRY=text	Optional, used with the LANGUAGE option above. A 2-character ISO 3166 country code - for example, US for the United States or JP for Japan.
DECODE=text	The format of the text if not plain ASCII. For example, use GBK, cp936 (Simplified), cp950 (Traditional), or euc-cn for Chinese, cp932 or euc-jp for Japanese, or cp949 or euc-kr for Korean if that is how the string is encoded.
URL=text	The URL for a web page to load rather than a link to a page in the current PDF document. You may also use "mailto:me@mysite.com" syntax to open an email window. Leave this option off for a standard outline entry.
PRINT[=text]	The outline entry will be used as a print function rather than a link to a page in the current PDF document. Use PRINT by itself to simply print to the default printer. Use PRINT=Dialog to bring up the printer dialog box. Leave this option off for a standard outline entry.
RANGE=name	Optional. Sets the name of the page range to use with the PRINT option. Using PRINT without this option prints all pages. See the FROMPG and THRUPG commands for setting up a page range. The outline entry will print the page range to the user's default printer when clicked. There is no user intervention for printer selection or other options. The value is case sensitive.
SUB_LEVEL=number	If you are creating a sub-level along with this upper-level outline, enter the sub-level number (usually set to LEVEL + 1).
SUB_DESCR=text	The description that appears in the bookmarks pane for the sub-level item.
SUB_CLOSED	Adding this option will cause the initial display of the sub-level bookmark to be closed.
SUB_COLOR=color	The color to use for the sub-level entry (must be an RGB color). *(Acrobat 5.0 or above)*

Parameter	Description
SUB_ITALICS	Italics outline entry *(Acrobat 5.0 or above)*
SUB_BOLD	Boldface outline entry *(Acrobat 5.0 or above)*
SUB_URL=text	The URL for a web page to load rather than a link to a page in the current PDF document. You may also use "mailto:me@mysite.com" syntax to open an email window. Leave this option off for a standard outline entry.
SUB_PRINT[=text]	The outline entry will be used as a print function rather than a link to a page in the current PDF document. Use SUB_PRINT by itself to simply print to the default printer. Use SUB_PRINT=Dialog to bring up the printer dialog box. Leave this option off for a standard outline entry.
SUB_RANGE=name	Optional. Sets the name of the page range to use with the SUB_PRINT option. Using PRINT without this option prints all pages. See the FROMPG and THRUPG commands for setting up a page range. The outline entry will print the page range to the user's default printer when clicked. There is no user intervention for printer selection or other options. The value is case sensitive.

You should issue the command just after the <PAGE> command. The outline for this document is created as:

```
<PAGE>
<OUTLINE LEVEL=1 DESCR="Commands" CLOSED>
. . .
<PAGE>
<OUTLINE LEVEL=2 DESCR="Document Level" CLOSED SUB_LEVEL=3 SUB_DESCR="PDF">
. . .
<PAGE>
<OUTLINE LEVEL=3 DESCR="OUTLINE">
```

**<PAGELAYOUT
VALUE=text>**

Used to set the initial layout of the document when opened.

Parameter	Description	
VALUE	One of following values:	
	SinglePage	Display one page at a time (default)
	OneColumn	Display the pages in one column
	TwoColumnLeft	Display the pages in two columns,with odd-numbered pages on the left
	TwoColumnRight	Display the pages in two columns,with odd-numbered pages on the right

**<PAGEMODE
VALUE=text>**

Used to set the initial display of the document when opened.

Parameter	Description	
VALUE	One of following values:	
	UseNone	Neither document outline nor thumbnail images visible (default when not using an outline)
	UseOutlines	Document outline visible (default when using an outline)
	UseThumbs	Thumbnail images visible
	FullScreen	Full-screen mode with no menu bar, window controls or any other window visible

<PDF>

This is the opening tag to any report. You must issue this command and it must be the first command. You may optionally include a closing </PDF> tag at the end of your report.

The report will be interpreted as HTML if HTML rather than PDF appears as the first tag. Many of the tag options will not work correctly in this case. Report Writer will attempt to convert the HTML into PDF but keep in mind that is not what Report Writer is designed for so you may not get the desired result.

<PDFINCLUDE
SRC=text
PDFOVERLAY>

PDFINCLUDE is not available with PDF Report Writer SE.

Used to specify an existing PDF file to use pages from as backgrounds when the SRC option is used. This command is a short-cut for the PDFPAGE tag when the SRC option is left off of PDFPAGE. Therefore this command, if used, should only occur once in your document. You may combine pages from more than one PDF but you'll need to use the SRC option on PDFPAGE in that case. The command works the same as the -pdf command line option.

Also can be used to specify PDF overlays will take place with the background PDFs. This is to tell the software that pages from two or more included PDFs will overlap or appear together on the same output page. This will involve some extra processing on these included PDFs.

See the backgrounds section for more information on including PDFs. Also see the PDFPAGE command.

Parameter	Description
SRC=text	The name of the default PDF to use for backgrounds with the PDFPAGE command. This must be an existing PDF file. You may be able to use Excel/Word/PowerPoint files as well. See the PDFPAGE command for details on these file types.
PDFOVERLAY	Specifies that multiple background PDF pages will be used as a single background for a page. That is, two or more pages from the embedded PDFs will be used together as a background for a single output page. You do not have to specify a value for SRC with this option.

<PERL
INCLUDE>
</PERL>

Used as part of an XML data merge in the layout file. Place Perl code between the opening and closing tags when you want to send different Report Writer tags based on the input data. The names of the variables used in the XML data file may be used in the Perl code for conditional items. Use the same syntax for the fields as with the input layout file (tag with an & in front of the variable name).

Use the Perl "return" command to return to Report Writer the tags based on any conditional statements. All variables must be declared using the "my" or "local" Perl keywords as appropriate.

You must run Report Writer with the -allowperl option or setAllowPerl method or the Perl code will not be executed. Instead, the tag(s) between the <NOPERL> and </NOPERL> (if any) will be used. See the XML Data Merge section for more information.

Parameter	Description
INCLUDE	Use this option to include this PERL code block in code execution later in the report. For example, you might have a common subroutine you want to use in various locations. Use the INCLUDE option and place the subroutine in the PERL block. The code won't be executed right away but rather stored and used in all other PERL blocks where the INCLUDE option isn't specified.

**<PLAIN
VALUE=text>**

Used to apply the plain text parameter to all tables in the report. This option can be used to speed up processing at the expense of losing some formatting.

Parameter	Description
VALUE=text	An optional text character to use as the sizing character. The character "1" is used as the default.

Sizing variable width characters to determine fit in a table cell can take a significant amount of time in a large report. This option cuts down on the time needed by assuming all characters are a standard width. This allows the program to not have to check each letter to determine size. All tags (other than <BR>) are ignored in the text. Also, only left alignment is used.

The default character width is based on the number 1. You may set a different character (such as M if you need a wider character) with the VALUE option on this tag. Text flowing past a cell border means you probably need a wider character. All tables will use this option unless you set PLAIN="" in the TABLE tag. In addition, you may leave this global option out and use the PLAIN option in the TABLE tag itself.

This option should only be used when you need a quicker view of the data and are less concerned with appearance.

<SHOWERRORS>

Used to debug SQL or other errors that are sent to the error log file. Prints any errors at the end of the output PDF.

<QUICKBUILD>

Sets the quick build option. This is same as setting the -q option on the exe command line or using the setQuick method of the DLL. See the Build Options section for more information. Note this option builds the PDF in memory first before writing to the output file.

<QUICKBUILD2>

Sets the quick build option. This is same as setting the -q2 option on the exe command line or using the setQuick2 method of the DLL. See the Build Options section for more information. Note this option builds the PDF to disk (when an output file is used) as it goes rather than in memory like <QUICKBUILD> It's best to use this option over <QUICKBUILD> unless you are using the &totpage variable.

<RTFIMAGES
NOCONV
TYPE="text">

Images are not exported by default when creating RTF documents. Include this tag to export all images within text/table tags. This will, of course, increase the size of the RTF document. Not all RTF readers (such as WordPad) will be able to view the images.

Use the -rtf option on the command line option to create RTF documents.

Parameter	Description
NOCONV	Specify this option only when your image is not working properly by default. This option will bypass some of the internal conversions which are usually necessary but may cause problems with certain high-resolution images. All images must be either jpeg or png format.
TYPE="text"	Specify this option only when your image is not working properly by default. Currently, only "jpeg" is valid. This is the format to convert the image to, not the source format of the image.

**<RTFLINESPACE
VALUE=number>**

Used to modify the amount of spacing between lines when exporting to RTF. This option should only be used if you need to fine tune the amount of line spacing you're getting the RTF. The default amount should suffice for most situations. Negative numbers will add more spacing while positive numbers reduce spacing.

Parameter	Description
VALUE=number	An amount to adjust the line spacing.

**<RWPATH
DIR=text>**

Used to supply a base directory for images, fonts and other external files. You may include this tag multiple times with a different directory setting for each one. The directories will be searched in order whenever a file cannot be found.

For example:

```
<PDF>
<RWPATH DIR="c:\files\dev">
<RWPATH DIR="c:\files\test">
<RWPATH DIR="c:\files\prod">
<PAGE>
<TEXT>
If the image is not found in the relative location
the path settings will then be used to locate it.
<IMG SRC="images\mypic.jpg">
</TEXT>
```

Report Writer will search for the image in the relative location first ("images\mypic.jpg"). If the image is not found, it will start by looking in c:\files\dev\images. If the image cannot be found, it will then look at c:\files\test\images and finally in prod. Not all of the directories have to exist. At least one should be valid in order to find the image, however.

You may also do the same with other files, such as fonts with the ADDFONT command or an XML data file. The directories will be searched in the same manner.

Parameter	Description
DIR=text	The base directory. For example, "c:\myfiles\dev" or "/files/dev/". The ending slash is optional.

<SET
NAME=text
PAGE=text>
</SET>

Used to create a variable or function that performs some text or color settings. You then refer to the setting by using an & character with the value used for NAME in the SET statement. Place any statements between the opening and closing SET statements you wish to reference later in a table or text block. You can use this option to set the format for some text or store a frequently used text string. You can also, optionally, pass parameters into the function. Parameters are passed by reference and are referred to in the SET block as $1 for the first variable, $2 for the second, etc. Use parenthesis ( ) around variables when passing them to the function. For example:

&myfunct("Parameter 1","Parameter number 2");

Parameter	Description
NAME=text	The variable name to use for the block of code that follows. Names are case sensitive.
PAGE=text	Optional. Used to specify that the SET statement should only apply to the pages in the named range. Use the FROMPG/THRUPG tags to name a range. This allows you to specify different values for the same variable in different areas of the report.
UTF-8	Optional. Used to denote the text contains one or more UTF-8 characters.

The following is one example of how to use the SET command.

```
<PDF>

<SET NAME=client>
<FCOLOR VALUE=blue>
<FONT FACE=16 SIZE=12>Sample Company</FONT>
<FCOLOR VALUE=black>
</SET>

<SET NAME=hdr>
<BR><LINE><BR VALUE=.5> <FCOLOR VALUE=red>
<FONT FACE=16 SIZE=18>$1</FONT>
<FCOLOR VALUE=black><BR>
</SET>

<PAGE>
<TEXT>
Client: &client
```

```
&hdr("Section 1");
Here is some text for section 1.
&hdr("Section 2");
Here is some text for section 2.
</TEXT>
```

<SETPG
VALUE=number>

Used to set the page number (see the page variable). If you have a report with multiple sections and you want to set the page number to 1 at the beginning of each section, issue this command before the <PAGE> command.

Parameter	Description
VALUE=number	The new page number to start numbering with.

```
<SHADING
    NAME=text
    COLOR1=color
    COLOR2=color
    COLOR3=color
    COLOR4=color
    COLOR5=color
    COLORARY=text>
```

Used to define a gradient shading pattern. The shading pattern can then be used for table cell, text, rectangle or page backgrounds. You may specify from two to five colors. See an example here.

Parameter	Description
NAME=text	The name for the shading pattern. Must be unique within the document.
COLOR1=color	The starting color. Any valid color code.
COLOR2=color	Any valid color code.
COLOR3=color	Optional. Any valid color code.
COLOR4=color	Optional. Any valid color code.
COLOR5=color	Optional. Any valid color code.
COLORARY=text	A comma separated list of 4 or 6 numbers. The default is 0,0,1,0. These represent the X_0, Y_0, X_1, Y_1 matrix coordinates for the shading pattern. A matrix of 0,0,1,0 goes from left to right from COLOR1 to COLOR_n. A matrix of 0,0,0,1 goes from top to bottom. You may use decimals or negative numbers as well to offset where the middle of the gradient lies. Use 6 numbers [x0, y0, r0, x1, y1, r1] for a radial type shading, such as for a sphere. The numbers specify the centers and radii of the starting and ending circles, expressed in points. The radii r0 and r1 must both be greater than or equal to 0. If one radius is 0, the corresponding circle is treated as a point; if both are 0, nothing is painted.

<SIZEHEADINGS>

Forces previous TR rows with SUBHEADING specified to be used in computing column widths for a MODTABLE tag. Typically, these rows are not repeated when computing the column width layout for data contained in a MODTABLE structure. Only useful when using the MODTABLE tag and TR subheadings that repeat across MODTABLE tags. See the TABLE tag for more details.

<SOFTHYPHEN
VALUE=text>

Sets the character to use as the soft hyphen. The default is ASCII code 173. This should only be used when the default does not work with any added fonts.

Parameter	Description
VALUE=text	A single character to use as the soft hyphen. Typically you'll want to use the dash or minus sign character.

<STARTPG
VALUE=number>

Used to set the page number to open the PDF to. Normally the first page is the initial page but this tag allows you to override.

Parameter	Description
VALUE	The page to open the document at.

<STYLE>
</STYLE>

Used to place style sheet information in the document. The style sheet data goes between the opening and closing STYLE tag. It's usually better to keep the style sheet in a separate file and refer to it using the LINK tag though. That way the style information is kept separate from the PDF commands and can be modified without changes to the process that creates the Report Writer commands.

See the LINK tag for more information on setting up a style sheet for use with Report Writer.

<SUBJECT
LANGUAGE=text
COUNTRY=text>
</SUBJECT>

Sets the text for subject of the document. Place the text between the opening and closing tags.

Parameter	Description
LANGUAGE=text	Text is assumed to be Unicode (2-character format) when this is used. A 2-character ISO 639 language code - for example, EN for English or JA for Japanese. The complete list of codes are available through http://www.iso.ch.
COUNTRY=text	Optional, used with the LANGUAGE option above. A 2-character ISO 3166 country code - for example, US for the United States or JP for Japan.

<TABLESIZE>
</TABLESIZE>

-- or --

<TABLESIZE
LINES=number>

Sets the table sizing section. This tag is used to increase performance by marking a section at the top of the report as the only area which should be examined during the table column sizing process. Report Writer normally scans the entire input file looking for all tables and the data contained in the columns. This is necessary to size the table columns to their optimal size. By placing this information within these tags near the top of the report, only this subset of the input need be searched for this information. This can increase performance, especially when there are one or two large tables making up the bulk of the report. While this option does allow for faster build times, you might not get optimal table column widths since not all of the table rows are examined. The EXCEL command will not work with this command since the data for Excel is captured during this pass of the file. Also see the INITIALIZE option.

There are two ways to use this tag. The first is to place the table sizing information between opening and closing <TABLESIZE> and </TABLESIZE> tags. This allows you to place a sample row from all tables in this section. All information placed here, such as text, tables and images, are ignored when the output is built. This section is only used for table column sizing. You must use the NAME option on all tables in the report when using the tags this way. The NAMEs in the TABLESIZING section must match the NAMEs used in the report itself. Place one or more sample row from each table in this section. Be sure to have a PAGE command and, if necessary, a GRID command just as is used in the report where the table is actually placed.

An alternate method is to use the TABLESIZE tag with the LINES option instead. There is no closing tag in this instance and, if a closing tag is found, it overrides the LINES option. The LINES option tells the program to look at the next X number of lines in the input file or set of commands. Only the following X number of lines are examined on the column sizing pass so all tables used in the report must fall within that section. Also, unlike the above method, all information is included as part of the report. This feature works best when you have a single large table making up most of the report.

See the Build Options section for more information.

<TEXTABS>

Sets the ABS parameter throughout the document on all TEXT commands. This affects where text is placed when centered or right aligned. See the TEXT command for more information.

**<THRUPG
NAME=text>**

Used with the button input statement to specify the end page for printing. This tag is used in conjunction with the FROMPG tag to mark a page range for printing. You may use a closing </FROMPG> tag instead of THRUPG if there is no overlap of page ranges.

You may also use this tag to mark a range of pages for retrieval with the GetPageRange DLL method or for use with the OUTLINE tag.

Parameter	Description
NAME=text	The name of this page range. Place this tag where you want to mark the point to end the range. There should exist a corresponding FROMPG tag somewhere in the document with the same value. Name values are case sensitive.

<TITLE
VIEWERTITLE
LANGUAGE=text
COUNTRY=text>
</TITLE>

Sets the text for title of the document. Place the text between the opening and closing tags.

Parameter	Description
VIEWERTITLE	Use the document title in the title bar of the viewer. The file name of the PDF is used if this is not set. *(Acrobat 5.0 or above)*
LANGUAGE=text	Text is assumed to be Unicode (2-character format) when this is used. A 2-character ISO 639 language code - for example, EN for English or JA for Japanese. The complete list of codes are available through http://www.iso.ch.
COUNTRY=text	Optional, used with the LANGUAGE option above. A 2-character ISO 3166 country code - for example, US for the United States or JP for Japan.

<TRANSPARENCY
NAME=text
VALUE=number
TRANSPMODE=text>

Used to define a transparency. A transparency is used in charts to allow a background image to show through on the chart at a desired transparency. Create one more transparencies then refer to them by the same name in the chart. Note that Acrobat and Reader may darken the contents of the pages where the transparency is used. This is because of the way Adobe handles the transparency by converting color spaces from RGB to CMYK.

Parameter	Description
NAME=text	The name you want to use for the transparency.
VALUE=number	The percentage of transparency. Enter a value from 1 to 100. Smaller values make the area more transparent - that is, allows more of the background to show through.
TRANSPMODE=text	The mode for the transparency (optional). Each provides a different effect for the overlay. The valid values are: Normal (Default) Multiply Screen Overlay Darken Lighten ColorDodge ColorBurn HardLight SoftLight Difference Exclusion Hue Saturation Color Luminosity

<URL
    ADDRESS="text"
    PARAMS="text"
>

This tag sends a post containing the parameters you specify once a PDF is finished building. Use it to create an alert, trigger an IFTTT recipe or other notification from your website.

Parameter	Description
ADDRESS="text"	The http or https address you want to use. Any errors will show up in the build log (setBuildLog method or -buildlog option).
PARAMS="text"	Optional. Any JSON formatted parameters as a string. For example, "{'status':'done', 'pages':121s}". Use the variable 121s for the number of pages. Use the variable &pdf for the PDF output (converted to Base64).
METHOD="text"	Optional. Default is GET. For example, METHOD="POST".

<VALIGN2>

Used to set an alternate method of vertical alignment. This method can be used with ROWSPANned cells but you cannot use QUICKBUILD or QUICKBUILD2 with it. The default method of vertical alignment can be used with QUICKBUILD or QUICKBUILD2 but will only work in a table cell that is not part of a rowspan.

Only use this when you require vertical alignment within a cell using the ROWSPAN option.

<ZOOM
VALUE=number|FITPAGE|FITWIDTH>

Used to set the initial zoom factor. Default is dependant on user settings.

Parameter	Description
VALUE	The zoom factor to open the document at. Enter 100 for 100 percent. FITPAGE = open the document sized so the entire page fits in the window. FITWIDTH = open the document sized so the width of the page fits in the window.

Page Level

Page level commands are used to start a page or assign certain settings such as headers or footers. A valid Report Writer input stream or file must have at least one <PAGE> command. All other commands in this section are optional.

<BODY
LINK=color>
</BODY>

Encloses the body text and/or tables. Remember to include this tag when placing content on a page if you have headers and/or footers. If you leave it out your text might overlay your header/footer.

While inside of the BODY tag, the amount of space for YUNITS as defined by the GRID is adjusted to be the area within the BODY rather than the page. For instance, say you are using an 8.5 x 11 piece of paper and have it divided into 100 units from top to bottom. You really have 8 inches along the Y axis rather than 11 if the header takes an inch and the footer takes two inches. Since the YUNITS are adjusted, then a value of 0 means 1 inch down and value of 100 is 9 inches from the top (or 8 inches from the bottom of the header).

Let's say you are also drawing a box around the body - from Y position 0 to 100. Now it doesn't matter if you decide to add another line in the header or footer, the box will still ecompass the body within the header and footer.

Parameter	Description
LINK=color	Optional - sets the text color for links.

<CLEAR
HEADER
SUBHEADER
FOOTER
SUBFOOTER>

Used to clear headers/footers set with the HEADER, SUBHEADER, FOOTER, and SUBFOOTER tags. This command may be issued anywhere on the page. Alternatively, you may start a new page and issue a header or footer command with the new information if you wish to modify.

Parameter	Description
HEADER	Clears the page HEADER.
SUBHEADER	Clears the page SUBHEADER.
FOOTER	Clears the page FOOTER.
SUBFOOTER	Clears the page SUBFOOTER.

```
<FOOTER
    MARGIN=units
    IFEVEN
    IFODD
    PAGE2
    PAGELAST>
</FOOTER>
```

Encloses the footer information.

Parameter	Description
MARGIN=units	Sets the margin of space in user defined units based on the GRID command.
IFEVEN	Specifies that this footer is for even numbered pages.
IFODD	Specifies that this footer is for odd numbered pages.
PAGE2	Specifies that this footer replaces the current footer after the first page. This allows you to specify one footer for the first page and a different footer for the rest of the pages. In this case you would have two FOOTER blocks and have this option set on one of them.
PAGELAST	Specifies that this footer replaces the current footer on the last page where the current BODY tag is closed (may or may not be the last page of your PDF however). The page space reserved for the footer on all pages in the BODY section will be the larger of the current footer and this final footer. See the PAGEFOOTER tag as well.

Between the <FOOTER> and </FOOTER> tags you place content such as text and/or tables. The size needed for the footer will be determined by the program. When you place information in the BODY section the program will automatically place that information above the footer.

```
<GRID
    ML=inches
    MR=inches
    MT=inches
    MB=inches
    XUNITS=number
    YUNITS=number>
```

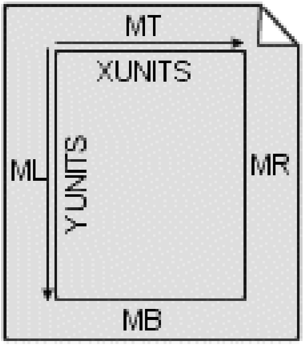
Sets the margins and virtual grid for each page. The default margins are 1/2 inch all around if you do use a grid command. All commands that use X and Y coordinate values are based on the grid values for XUNITS and YUNITS. For instance, with any size page using equal margins left/right and top/bottom and with GRID settings of XUNITS=50 and YUNITS=100, the middle of the page is at X=25 and Y=50.

Note you may not change the grid layout once inside a HEADER, FOOTER or BODY section of the report. You should ideally set the GRID once before issuing your first page command and use that setting throughout the report.

Parameter	Description

ML=inches
MR=inches
MT=inches
MB=inches
	Sets the various margins for a page of any size. If the METRIC command is used, enter the number of centimeters instead.
	ML = Margin Left
	MR = Margin Right
	MT = Margin Top
	MB = Margin Bottom

XUNITS=number	User defined - number of columns to divide the printable area (page width - left/right margin) into.
YUNITS=number	User defined - number of rows to divide the printable area (page height - top/bottom margin) into.



Do not confuse the rows and columns of XUNITS and YUNITS with table rows and columns. These are simply for convenience when placing information on a page. The YUNITS value is not as important as the XUNITS value since the program will take care of content flow. If you are drawing a rectangle on the page then the YUNITS may be important as the corners of the box will be specified in XUNITS and YUNITS.

The Y coordinate is always 0 for the top line when inside of a TEXT block. This will allow you to move to the top of the text block without having to know where the text block is positioned on the page. For example, the footer of this document contains some text left justified and the page number right justified. This was done by first printing the left portion, then specifing a value of 0 for Y position, setting a right alignment, then printing the page number.

The XUNITS is a little more important as you will use these corrdinates to set your left and right margins for text blocks and/or tables. The TEXT and TABLE commands will simply use a value of 0 for the left margin and the value entered for XUNITS for the right margin if none are supplied.

```
<HEADER
    MARGIN=units
    IFEVEN
    IFODD
    PAGE2>
</HEADER>
```

Encloses the header information.

Parameter	Description
MARGIN=units	Sets the margin of space in user defined units based on the GRID command.
IFEVEN	Specifies that this header is for even numbered pages.
IFODD	Specifies that this header is for odd numbered pages.
PAGE2	Specifies that this header replaces the current header after the first page. This allows you to specify one header for the first page and a different header for the rest of the pages. In this case you would have two HEADER blocks and have this option set on one of them.

Between the <HEADER> and </HEADER> tags you place content such as text and/or tables. The size needed for the header will be determined by the program. When you place information in the BODY section the program will automatically place that information under the header.

```
<MATRIX
    ANGLE=number
    A=number
    B=number
    C=number
    D=number
    XOFF=number
    YOFF=number
    X=number
    Y=number>
</MATRIX>
```

Used to angle or rotate the contents that follow. This differs from the ROTATE command in that everything is rotated with MATRIX rather than just text. No page breaks will automatically occur while a MATRIX operation is in effect. You may also use this option to scale (set A and D to decimal values other than 1) or skew (set B and C to values other than 0) contents as well.

Parameter	Description
ANGLE=number	Specify ANGLE (in the range from 0 to 360) to compute the A, B, C, and D values automatically. The rotation is around the lower left point of the page. The formula is a matrix computation where: x' = x*a + y*c + xoff y' = x*b + y*d + yoff where: a = cos(angle) b = sin(angle) c = 0 - b d = a The angle of rotation is counter clockwise. Unless X and Y are specified, the rotation occurs around the lower left corner of the page.
A=number	The value to use for A. The default is 1.
B=number	The value to use for B. The default is 0.
C=number	The value to use for C. The default is 0.
D=number	The value to use for D. The default is 1.
XOFF=number	A positive or negative offset amount in points (1/72 of an inch) to add the new X value.
YOFF=number	A positive or negative offset amount in points (1/72 of an inch) to add the new Y value.

Parameter	Description
X=number	The X grid position to use for the point of rotation. If X or Y are not specified, the lower left corner of the page is used.
Y=number	The Y grid position to use for the point of rotation. If X or Y are not specified, the lower left corner of the page is used.

Page Level Commands

```
<PAGE
    HEIGHT=inches
    WIDTH=inches
    TYPE=text
    ORIENT=landscape|portrait
    NEWFORM=text
    STARTSUB
    PGNOSTYLE=text
    PGNOPREFIX=text
    PGNOSTART=number
    BGCOLOR=color
    BORDER=number
    BORDERCOLOR=color
    BORDERMARGIN=number
    TRANS=text
    SECS=number
    TDIR=H|V
    MDIR=I|O
    JSOPEN=text
    JSCLOSE=text
    NOBREAK=number
    CLEARFORMS
    YPOS=number
    IFEVEN
    IFODD
    ROTATE=number
    TRANSPARENCY=number
    TRANSPMODE=text
    SCALE=number
    SCALEX=number
    SCALEY=number
    EVENRIGHT=number
    ODDRIGHT=number
    EVENUP=number
    ODDUP=number
    SHRINKPAGE
    SHRINKROUND=number
    SHRINKVALIGN=text
    ENDX|ENDLINE|ENDSTR=text
    ENDFACE=number
    ENDSIZE=number
    ENDFCOLOR=color
    ENDSCOLOR=color
    ENDPAGE>
```

Starts a new page. All of the parameters are optional. If you are doing a page break and you are changing the width or height, be sure to close out any open BODY tags and reissue new HEADER and/or FOOTER commands.

Parameter	Description
WIDTH=inches	Sets width of page in inches. Default is 8.5. If the METRIC command is used, enter the number of centimeters instead.
HEIGHT=inches	Sets height of page in inches. Default is 11.
TYPE=text	The page size to use if not using Width or Height options. See the following table for a list of page types.
ORIENT=landscape|portrait	Landscape sets page size to 11 by 8.5 if you leave out height and width. Portrait sets page size to 8.5 by 11. If height and width have been set differently by a previous page command, setting ORIENT to landscape will set the width to the larger of width and height while portrait will set the width to the smaller of the two.
NEWFORM=text	To use this page as a background for other pages, enter a value for NEWFORM. This page will not print but may be included on other pages with the USEFORM tag.
STARTSUB	Starts a new sub-section. This is only for the &pagesub; and &totpagesub; variables. The &pagesub; will be set to 1 and &totpagesub; will be set to the total number of pages in this sub-section. The sub-section ends when a new PAGE tag is encountered with STARTSUB set or when the end of the report is reached.

Page Level Commands

Parameter	Description

PGNOSTYLE=text	This option allows you to specify a numbering style for a set of pages. Note this is for the viewer application only and does not affect the &page variable. A single letter code (case-sensitive). Default is D. Values are:
	D = Decimal arabic numerals
	R = Uppercase roman numerals
	r = Lowercase roman numerals
	A = Uppercase letters
	a = Lowercase letters

PGNOPREFIX=text	An optional prefix (ex. A-) to use in front of the page number. This option is only for the viewer application. Setting to A- will cause the pages to be numbered A-1, A-2, etc.
PGNOSTART=number	The starting page number for display. This option is only for the viewer application. For example, if you have 3 pages in the front of the document numbered in roman numerals and you want to start arabic numbering on page 4 (but have page 4 show as 1, 5 as 2, etc.), you'd set this value to 1.
BGCOLOR=color	Sets the color of the page background. Default is white. Use BGCOLOR="" to set the background back to white.
BORDER=number	Sets the border width for the page. Default is 0, or no border. The value is the thickness of the border in points (1/72 of an inch).
BORDERCOLOR=color	Sets the color for the page border. Default is black.
BORDERMARGIN=number	Margin for the page border in points (1/72 of an inch). By default the border is flush against the edge of the page. This option moves the border in - away from all edges.

Parameter	Description
TRANS=text	One of the following transition types: Split Blinds Box Wipe Dissolve Glitter The default page transition is to just replace the current page with the new page. The effects listed above can be used for presentation purposes.
SECS=number	The length of time in seconds to spend on the page transition. Default is 1 if a transition type is used.
TDIR=H|V	The direction of the transition, horizontal or vertical. Valid only for Split and Blinds transition types.
MDIR=I|O	The direction of motion for the transition effect, inward or outward from the center of the page. Valid only for Split and Box transition types.
JSOPEN=text	JavaScript to execute when the page is opened.
JSCLOSE=text	JavaScript to execute when the page is closed.

Parameter	Description
NOBREAK=number	Page breaks will occur by default when a table or block of text hits the bottom of the current page. Set NOBREAK to a value of 1 or 2 to prevent this behavior. A value of 0 means turns off the NOBREAK option (if you're in the BODY of a document and you specified a NOBREAK prior to this page). A value of 1 means continue printing text beyond the page margin or even off the page. A value of 2 means cut the text or table at the margin.
CLEARFORMS	Clear any USEFORM statements that had been issued. The contents of a USEFORM background are used for all subsequent pages which are in the BODY of the document. This option clears out any USEFORM statements so the background no long prints. This allows you to remove the background or specify a new one.
YPOS=number	Conditionally page break based upon the current Y position. For example, if the current GRID settings are 100 in the Y direction and you set YPOS=70, the PAGE command will be ignored if the current Y position is less than 70. You may also enter the value followed by a % sign (i.e. YPOS=65%) to denote a percentage of the current GRID.
IFEVEN	Page break only if the current page is an even numbered page. Page numbering starts at 1 for the first page in the document. If the current page is an odd numbered page, the PAGE command is ignored.

Parameter	Description
IFODD	Page break only if the current page is an odd numbered page. Page numbering starts at 1 for the first page in the document. If the current page is an even numbered page, the PAGE command is ignored.
ROTATE=number	Rotation angle to set the page at when viewing. Valid values are 90, 180 or 270.
OPACITY=number	The opacity for the page when using a background (see the PDFPAGE command). Set to a value between 1 and 100. The default is 100 where the page contents are opaque - that is, nothing from the background shows through in areas where there are text or graphics in the top level. Opacity or transparency is used in Acrobat and Adobe Reader 5.0 and higher.
TRANSPMODE=text	The mode for the transparency (when using the OPACITY option). Each provides a different effect for the overlay. The valid values are: Normal (Default) Multiply Screen Overlay Darken Lighten ColorDodge ColorBurn HardLight SoftLight Difference Exclusion Hue Saturation Color Luminosity

Page Level Commands

Parameter	Description
SCALE=number	Percentage to scale the page contents by (both horizontally and vertically). The physical page size remains the same, only the contents are scaled. For example, use 80 for 80%.
SCALEX=number	Percentage to scale the page contents by horizontally. For example, use 80 for 80%. Setting the SCALE parameter overrides this setting.
SCALEY=number	Percentage to scale the page contents by vertically. For example, use 80 for 80%. Setting the SCALE parameter overrides this setting.
EVENRIGHT=number	Value in points (1/72 of an inch) to move even numbered page contents to the right. May be a positive or negative value. Negative values move contents to the left. Page numbering starts at 1 for the first page in the document.
ODDRIGHT=number	Value in points (1/72 of an inch) to move odd numbered page contents to the right. May be a positive or negative value. Negative values move contents to the left. Page numbering starts at 1 for the first page in the document.
EVENUP=number	Value in points (1/72 of an inch) to move even numbered page contents up. May be a positive or negative value. Negative values move contents down. Page numbering starts at 1 for the first page in the document.

Parameter	Description
ODDUP=number	Value in points (1/72 of an inch) to move odd numbered page contents up. May be a positive or negative value. Negative values move contents down. Page numbering starts at 1 for the first page in the document.
SHRINKPAGE[=number]	Makes the height of the page equal to the last Y position where a text command ended. The bottom margin specified by the GRID command is included when the page is shortened. Do not use this option within a BODY section of a report. Optionally set to the number of points (1/72 of an inch) additional to drop down.
SHRINKROUND=number	Optional - for use with SHRINKPAGE. The number of units to round up to when resizing the page. The value is in inches (centimeters when METRIC is used). Setting to a value of 2, for example, rounds the page size up to the nearest even number of inches.
SHRINKVALIGN=text	Optional - for use with SHRINKPAGE and SHRINKROUND. Vertically aligns the page contents when using SHRINKROUND. Set to "M" or "Middle" for middle of page or "B" or "Bottom" for bottom. Default is the top of page.

Parameter	Description
ENDX|ENDLINE|ENDSTR=text	These options are to provide some security against changes to a document by filling in white space with characters or lines. Use ENDLINE to draw a line from the upper left to the lower right corner of any remaining white space on the page. Use ENDX to draw another line from the upper right to the lower left corner, forming an X. Use ENDSTR=text to set a character or string to fill in the remaining white space.
ENDFACE=number	A font number to use for ENDSTR=text option.
ENDSIZE=number	The font size to use for the ENDSTR=text option.
ENDFCOLOR=color	The fill color to use to for the text.
ENDSCOLOR=color	The stroke color to use for the line or text.
ENDPAGE	Use the end page width and length rather than the margins.

There is no closing tag for <PAGE>. Simply issue another PAGE tag to start another page. Each report must include at least one page tag.

The following table lists the various built-in page sizes available for the TYPE option.

Type	Description
Standard	Standard, 8 1/2 x 11 in.
Legal	Legal, 8 1/2 x 14 in.
Letter	Letter, 8 1/2 x 11 in.
LetterSmall	Letter Small, 8 1/2 x 11 in.
Tabloid	Tabloid, 11 x 17 in.
Ledger	Ledger, 17 x 11 in.
Statement	Statement, 5 1/2 x 8 1/2 in.
Executive	Executive, 7 1/2 x 10 1/2 in.
A3	A3, 297 x 420 mm
A4	A4, 210 x 297 mm
A4Small	A4 Small, 210 x 297 mm

Type	Description
A5	A5, 148 x 210 mm
B4	B4, 250 x 354 mm
B5	B5, 182 x 257 mm
Folio	Folio, 8 1/2 x 13 in.
Quarto	Quarto, 215 x 275 mm
10x14	10 x 14 in.
11x17	11 x 17 in.
Note	Note, 8 1/2 x 11 in.
Env9	Envelope #9, 3 7/8 x 8 7/8 in.
Env10	Envelope #10, 4 1/8 x 9 1/2 in.
Env11	Envelope #11, 4 1/2 x 10 3/8 in.
Env12	Envelope #12, 4 1/2 x 11 in.
Env14	Envelope #14, 5 x 11 1/2 in.
EnvDL	Envelope DL, 110 x 220 mm
EnvC3	Envelope C3, 324 x 458 mm
EnvC4	Envelope C4, 229 x 324 mm
EnvC5	Envelope C5, 162 x 229 mm
EnvC6	Envelope C6, 114 x 162 mm
EnvC65	Envelope C65, 114 x 229 mm
EnvB4	Envelope B4, 250 x 353 mm
EnvB5	Envelope B5, 176 x 250 mm
EnvB6	Envelope B6, 176 x 125 mm
EnvItaly	Envelope, 110 x 230 mm
EnvMonarch	Envelope Monarch, 3 7/8 x 7 1/2 in.
EnvPersonal	Envelope, 3 5/8 x 6 1/2 in.
FanfoldUS	U.S. Standard Fanfold, 14 7/8 x 11 in.
FanfoldStdGerman	German Standard Fanfold, 8 1/2 x 12 in.
FanfoldLglGerman	German Legal Fanfold, 8 1/2 x 13 in.

<PDFPAGE
    SRC=text
    SRC2=text
    PDFOVERLAY
    ALLPAGES|VALUE=number
    VALUE2=number
    INSERTATPAGE=number
    USEFORM=text
    NOFIRSTBREAK
    SHOWONCE>

PDFPAGE is not available with PDF Report Writer SE.

Includes the page specified from the source PDF (or Excel/Word/PowerPoint document - see details below) as a background for the current page. The page will continue to be used as the background page when page breaks occur. The background page will be reset if you issue your own PAGE command or, if you're in the BODY of a report, close and re-open the BODY. In that case, be sure to reset or change the background page otherwise you won't get a background. This command should appear just after the PAGE command on the page you want the background on.

Use the PDFINCLUDE tag with the PDFOVERLAY option if you will have more than one PDFPAGE command on any single output page. The PDFOVERLAY option instructs the software to do some extra processing on the background PDFs for compatibility. Note that not all PDFs can be processed properly depending on the type of compression they utilize. As long as you are not using more than one PDF page as a background to any output page you don't need to worry about this.

You may use Excel, Word, or PowerPoint files as well under one of the following two conditions. First, if you have Excel/Word 2007 or higher with the free Office "Save as PDF" add-on installed, the Windows version of PDF Report Writer can connect to Excel or Word to perform the conversion.

Second, OpenOffice can be used to convert these document types to PDF. Under Windows, PDF Report Writer will attempt to connect to OpenOffice using the Windows COM sub-system to perform the conversion. This means that the standard OpenOffice installation should be all you need on Windows so PDF Report Writer can convert Excel/Word documents. The command line utility "unoconv" (part of OpenOffice) can also be used on any operating system if you have OpenOffice installed as a server. Be sure the PATH environment variable for the user running PDF Report Writer contains the location where unoconv is located. This is useful on Linux systems to convert Excel/Word to PDF. See the OpenOffice documentation for instructions on setting up the OpenOffice server.

While you can manually use one of these options to convert to PDF first then pass the PDF to PDF Report Writer, it may be more convenient to let PDF Report Writer run this process for you by simply passing your Office document to PDF Report Writer. Users do not need anything special on their local machine to read or convert Excel/Word to PDF as the process will be handled by the PDF Report Writer server with one of these options setup on a server running PDF Report Writer in Client-Server TCP/IP mode.

See the introduction section for the parameters or methods for specifying an existing PDF to use pages from. See the backgrounds section for more information on using backgrounds. See the XPSPAGE tag for backgrounds when creating XPS output.

Parameter	Description
SRC=text	The source PDF (or Excel/Word/PowerPoint) file to use. You may combine pages from multiple PDFs. You may also use the PDFINCLUDE command to globally set the SRC if you are working mainly with one PDF. Any PDFPAGE tags without a SRC option will then default to the PDF specified by the PDFINCLUDE command.
SRC2=text	The source PDF file to use with the VALUE2 option. The PDF file specified by the SRC option is used if this option is omitted.
PDFOVERLAY	Supply this option if your output is not rendering correctly for the background PDF and you are using more than one background PDF. This option will force additional processing to attempt to correct the issue. May not correct all issues with background PDFs.

Page Level Commands

Parameter	Description
ALLPAGES|VALUE=number	Use the VALUE=number option to specify the page number from the existing PDF to use as the background. Or, use ALLPAGES to pull in all pages from the existing PDF. A page break will be inserted in front of each page pulled so if you have a PAGE command just before this one, you'll wind up with an extra blank page. Use the NOFIRSTBREAK option to prevent the initial page break in this case. You'll probably want to use the SHOWONCE option as well if you're using the ALLPAGES option.
VALUE2=number	The background page number to use after a page break while in the body of a document. This allows you to switch to a different form once a page break occurs. Leave this option off to continue to use the page number set from the VALUE option on subsequent pages. The background PDF used is taken from the SRC2 option, if specified. The PDF from the SRC option is used if SRC2 is not specified.

Parameter	Description
INSERTATPAGE=number	Adds the background page at the specified page number in the document. Use this option when Report Writer is controlling page breaks, during a long table for instance, and you do not have control over the page break at the required point in the document. For example, you may have some standard disclaimer that should always go at page 2 in your output. In this case, set INSERTATPAGE=2. The VALUE option should be set to the page number from the background PDF you want to insert as page 2. Note that the background page is only added if the PDF has at least the number of pages - 1 specified by this option. If you have INSERTATPAGE=3 and your PDF has only 2 pages, the third page with the background will be added. On the other hand, if you only have one page and you specify this background should be page 3 then no background will be added. Your PDF will contain only the one page in that case.
USEFORM=text	Allows you to specify a previously defined Report Writer background page to place on this page. Enter the code used on the NEWFORM parameter of the page you want to overlay. This option is for when you are using the INSERTATPAGE option and wish to place other text or table data on top of this static background page.

Page Level Commands

Parameter	Description
NOFIRSTBREAK	Use along with the ALLPAGES option to specify you don't want the initial page break inserted automatically. This allows you to put in your own page break along with an OUTLINE command, for example, to go along with the first page of the included PDF. The PAGE command and the OUTLINE command should both come before the PDFPAGE command in this case.
SHOWONCE	The pulled PDF page (or pages) will be used as a background while in the BODY section of a report. The reason is to allow you to place text or tables on top of a PDF background that may span several pages. Use this option to prevent the pulled page from showing up on subsequent pages.

```
<CONDCMDS
     IFEVEN
     IFODD
     PDFOVERLAY
     MINPAGE=number>
</CONDCMDS>
```

Conditionally process the tags within the opening and closing CONDCMDS. For example, use the IFODD option if you only want to add a background PDF or some text if the current page in the output PDF is odd. Use MINPAGE=x to further refine the logic so only when the PDF has at least the number of pages specified are the commands within the CONDCMDS processed. If the conditions are not satisfied then any commands between the opening and closing CONDCMDS tags are ignored.

Parameter	Description
IFEVEN	Process the tags only if the current page is an even numbered page. Page numbering starts at 1 for the first page in the document. If the current page is an odd numbered page, the containing commands are ignored.
IFODD	Process the tags only if the current page is an odd numbered page. Page numbering starts at 1 for the first page in the document. If the current page is an even numbered page, the containing commands are ignored.
MINPAGE=number	Optional. The minimum number of pages the output PDF must have before including the commands.

<PULLPAGE
SRC=text
PAGE=number>

PULLPAGE is not available with PDF Report Writer SE.

Used to pull a page from an existing PDF. You should use this command right after a PAGE command. This command is intended to be used only with PDFs created by Report Writer. It may not work with other PDFs. Only the page commands are pulled - no embedded fonts, interactive widgets, links or annotations are pulled out. Use the PDFPAGE command for non-Report Writer PDFs. Any images used by the existing PDF are pulled and included in the new PDF. Be sure to include any embedded fonts in the main input file in the same order as used in the PDF you are pulling pages from. This command may appear as many times as necessary and each may reference a different PDF file.

Parameter	Description
SRC=text	The path and name of the existing PDF input file.
PAGE=number	The page number to pull out. Leave this option off to pull in all pages from the PDF.

<STRIPE
X1=number
Y1=number
X2=number
Y2=number
C1=color
C2=color
HEIGHT=number>

Use to draw a zebra stripe pattern on the page. The pattern will repeat on each page until a STRIPE tag is encountered with a HEIGHT=0.

Parameter	Description
X1=number	The X position in points (1/72 of an inch) from the left page edge to start at.
Y1=number	The Y position in points (1/72 of an inch) from the botom of the page to end at.
X2=number	The X position in points (1/72 of an inch) from the left page edge to end at.
Y2=number	The Y position in points (1/72 of an inch) from the bottom of the page to start at.
C1=color	Enter a valid color for the 1st stripe.
C2=color	Enter a valid color for the 2nd stripe.
HEIGHT=number	The height in points (1/72 of an inch) for each stripe.

<REPEAT
APPEND
CONTINUE=text>
</REPEAT>

Enclose text or tables that you want repeated for a block of text or a table that follows. This is different from a HEADER or SUBHEADER in that the HEADER and SUBHEADER only print at the top of a page. A REPEAT block can start anywhere on the page.

You may have some heading text or formatting that belongs with a table that follows but you don't want to place that text in the table itself. The reason would be that you don't want the table to expand as wide as the heading information is. Or you may have a heading that belongs to a group of tables and/or text. You use the REPEAT and /REPEAT tags to enclose the heading information you want repeated. Issue an empty set of REPEAT and /REPEAT tags to turn off the repeat section. The CONTINUE option is used to optionally specify some text that would print only after the first page break occurred.

Parameter	Description
APPEND	Use this option to append to the previous repeat information. Information in the current repeat block is cleared out if this parameter is missing.
CONTINUE=text	Set the text you want to print after the first page break. You reference this with the value &continue somewhere in the text that falls between the opening and closing REPEAT tags. Enclose the text in quotes.

Here is an example of how you would use a repeat block:

```
<REPEAT CONTINUE="(Continued)">
<LINE>
<TEXT>
Section I &continue - This text flows all the way across the page and will repeat
for each page the following two tables require.
Once the next set of opening/closing REPEAT tags are encountered, this text will stop repeating.
</TEXT>
<LINE>
</REPEAT>
<TABLE CELLPADDING=2 CELLXPADDING=2>
<TH><TD>Company</TD><TD>Amount</TD></TH>
<TR><TD>ABC Corp</TD><TD>100.00</TD></TR>
<TR><TD>Any Corp</TD><TD>150.00</TD></TR>
. . . (rest of the table) . . .
<TR><TD>ABC Corp</TD><TD>100.00</TD></TR>
<TR><TD>Any Corp</TD><TD>150.00</TD></TR>
```

```
</TABLE>
<TABLE CELLPADDING=2 CELLXPADDING=2>
<TH><TD>Returns</TD><TD>Amount</TD></TH>
<TR><TD>Product 1</TD><TD>-200.00</TD></TR>
<TR><TD>Product 2</TD><TD>-80.00</TD></TR>
. . . (rest of the table) . . .
<TR><TD>Product 79</TD><TD>-400.00</TD></TR>
<TR><TD>Product 80</TD><TD>-10.00</TD></TR>
</TABLE>
<REPEAT>
</REPEAT>
<TEXT>
Neither this text (nor the prior heading) will repeat for each page of this table.
</TEXT>
<TABLE CELLPADDING=2 CELLXPADDING=2>
<TH><TD>Company</TD><TD>Amount</TD></TH>
<TR><TD>ABC Corp</TD><TD>100.00</TD></TR>
<TR><TD>Any Corp</TD><TD>150.00</TD></TR>
. . . (rest of the table) . . .
<TR><TD>ABC Corp</TD><TD>100.00</TD></TR>
<TR><TD>Any Corp</TD><TD>150.00</TD></TR>
</TABLE>
```

```
<SUBFOOTER
    MARGIN=units
    IFEVEN
    IFODD
    CONTINUE=text>
</SUBFOOTER>
```

Encloses the subfooter information.

Parameter	Description
MARGIN=units	Sets the margin of space in user defined units based on the GRID command.
IFEVEN	Specifies that this subfooter is for even numbered pages.
IFODD	Specifies that this subfooter is for odd numbered pages.
CONTINUE=text	Set the text you want to print after the first page break. You reference this with the value &continue somewhere in the text that falls between the opening and closing SUBFOOTER tags. Enclose the text in quotes.

Subfooters may be changed throughout the body section of a document. The first subfooter should come just after the BODY tag. The information will repeat across page breaks until a new subfooter is issued. Issue subfooter tags only after a PAGE command from then on. The new subfooter will replace the previous one without affecting your FOOTER data. Send a <SUBFOOTER> followed by </SUBFOOTER> to clear out a subfooter.

<SUBFOOTERCONT
MARGIN=units
IFEVEN
IFODD
CLEAR>
</SUBFOOTERCONT>

Encloses the continued subfooter information.

Parameter	Description
MARGIN=units	Sets the margin of space in user defined units based on the GRID command.
IFEVEN	Specifies that this continued subfooter is for even numbered pages.
IFODD	Specifies that this continued subfooter is for odd numbered pages.
CLEAR	Use after the text or table to remove the continued message. You do not have to use a closing SUBFOOTERCONT tag in this case.

This tag is used to print a continued message on each page except the last of a report or section of a report. Place this tag after the SUBFOOTER tag (if using one - it's not necessary to have a SUBFOOTER tag along with this tag) and before any text or tables before. The contents of this section are printed on the first page (assuming the first page isn't the last page). The contents continue to print until a <SUBFOOTERCONT CLEAR> tag is encountered. The CLEAR option removes the text from the current page and stops the text from printing on any future pages. The option can be started up again by issuing a new SUBFOOTERCONT tag after a PAGE command (and SUBFOOTER if one is used).

Note that the <SUBFOOTERCONT CLEAR> can be placed anywhere in the report. It does not have to follow a PAGE command.

<SUBHEADER
MARGIN=units
IFEVEN
IFODD
CONTINUE=text>
</SUBHEADER>

Encloses the sub header information.

Parameter	Description
MARGIN=units	Sets the margin of space in user defined units based on the GRID command.
IFEVEN	Specifies that this subheader is for even numbered pages.
IFODD	Specifies that this subheader is for odd numbered pages.
CONTINUE=text	Set the text you want to print after the first page break. You reference this with the value &continue somewhere in the text that falls between the opening and closing SUBHEADER tags. Enclose the text in quotes.

Subheadings may be changed throughout the body section of a document. The first subheading should come just after the BODY tag. The information will repeat across page breaks until a new subheading is issued. Issue subheader tags only after a PAGE command from then on. The new subheading will replace the previous one without affecting your HEADER data. Send a <SUBHEADER> followed by </SUBHEADER> to clear out a subheading.

<USEFORM
VALUE=text
VALUE2=text>

Uses a page defined with the NEWFORM parameter as a background page. You may use multiple USEFORM tags to include more than one background.

See the CLEARFORMS option on the PAGE command for information on clearing out forms in the body of a document.

Parameter	Description
VALUE=text	The code used on the NEWFORM parameter of the page you want to use as the background.
VALUE2=text	The background code to use after a page break while in the body of a document. This allows you to switch to a different form once a page break occurs. This must be set to a valid NEWFORM code. Setting it to anything else will have the effect of removing the background after a page break.

<XPSFILE
FILE=text
OPEN>

XPSFILE is not available with PDF Report Writer SE.

Specifies the output XPS file to create similar to the -xps option or setXPSFile method. This tag is used to automatically create an XPS file without passing additional parameters on the command line.

See the XPS Document section for more information on XPS.

Parameter	Description
FILE=text	The output file to create. Be sure to fully qualify the path. If left blank, the name of the PDF output file will be used but with an XPS file extension instead of PDF.
OPEN	Automatically open the XPS file using the default XPS viewer.

<XPSKEEPNAMES>

XPSKEEPNAMES is not available with PDF Report Writer SE.

Prevents renaming of images or other resources from the background XPS file. Normally resources such as images or embedded fonts are renamed to prevent conflict when placing in the output XPS file. This option prevents that from happening.

This can be useful if you decide later to replace images in multiple XPS files that have been created. However, if you have more than one XPS background you are using and they share a common resource name, only one will be written to the output XPS file. This would result in an incorrect image or font being used. Do not use this option unless you are only using one background XPS or have no issues when using with multiple XPS backgrounds.

<XPSPAGE
SRC=text
VALUE=number
DOC=number
MATRIX=text
OPACITY=number>

XPSPAGE is not available with PDF Report Writer SE.

Includes the page specified from the source XPS as a background for the current page in the XPS file. This tag has no effect on the PDF output. The page will continue to be used as the background page when page breaks occur. The background page will be reset if you issue your own PAGE command. In that case, be sure to reset or change the background page otherwise you won't get a background. This command should appear just after the PAGE command on the page you want the background on.

If you have an existing PDF you want to use as an XPS background, the easiest way to convert it is to simply print the PDF as an XPS document. You'll need to install Microsoft's XPS Document Writer on XP based systems first. See the XPS Document section for more information on XPS.

Since XPS is a relatively new format and still evolving, not all XPS documents may work. You'll want to test yours to make sure it renders the way you want. XPS support and compatibility in PDF Report Writer will continue to be enhanced and become more robust over time.

Parameter	Description
SRC=text	The source XPS file to use.
VALUE=number	The page number of the XPS file to use as the background. The default is 1.
DOC=number	The document from the XPS file to use. The default is 1.
MATRIX=text	A transformation matrix to apply to the background. A set of six numbers separated by commas. The default is 1,0,0,1,0,0. The first 4 numbers are used for skew and rotation and the last 2 for x/y offset. The offsets are units of 1/96 of an inch. For example, to move the background to right 1 inch and up 2 inches, use 1,0,0,1,96,-192.

Parameter	Description
OPACITY=number	The opacity of the background. Set to a value between 0 and 100. The default is 100 where the contents are opaque - set to a lower number to reduce the intensity of the background. At 0 the contents are no longer visible.

Text

Text commands are used to place a block of text (which may contain images as well) on the page or apply some setting to the text. Text is drawn on the page by either a TEXT block or a TABLE. The tags in this section can be used to control how the text appears in terms of font, size and color.

<A
NAME=text|HREF=text|POPUP=text|EMBED=text
EMBED
MIME=text
ICON=text
FILENAME=text
SUBJECT=text
LINKLINE=number
LINKCOLOR=color
ANNOT=text
TITLE=text
COLOR=color
NORTF
LANGUAGE=text
COUNTRY=text>
</A>

Used to add a web link, anchor or annotation. See the LINKLINE and LINKCOLOR commands as well. Use the NAME to insert an anchor in the current document to link to. When you use the A tag this way you do not need a closing </A> tag. You may then reference that page by inserting an anchor using HREF and placing a # in front of the name.

Parameter	Description
NAME=text|HREF=text|POPUP=text|EMBED=text	Use one of the above in the anchor or link. NAME is used to name an anchor or POPUP. HREF is used to specify the location a link points to or for a highlighted popup note. POPUP is used to store the name of the annotation. EMBED is used to store an external file in the PDF. This can be any type of file such as a Word, Excel, CAD, movie or other file. The end user may open the attachment for viewing in the target application or save it to disk.

Parameter	Description
MIME=text	Optional. For use when EMBED is specified. The mime type associated with the embedded file. For example, application/vnd.ms-excel for Excel. Windows based versions of Report Writer will attempt to locate the correct mime type from the system registry when the PDF is built.
ICON=text	Optional. For use when EMBED is specified. The icon to use for an embedded file. The values are: Graph Paperclip Attachment Tag The paperclip is the default.
FILENAME=text	Optional. For use when EMBED is specified. The file name to use for the attachment. This is not the source file name on disk - it is the filename to refer to the embedded file by once it's in the PDF. The default is the name used in the EMBED entry.
SUBJECT=text	Optional. For use when EMBED is specified. The subject for the attachment.
LINKLINE=number	The width of the line to draw for the link. Used a value of 0 to turn off the line.
LINKCOLOR=color	The color of the text and line to draw for the link.
ANNOT=text	The text to place in an annotation. This is specified in the link using the NAME option. Display the annotation by using another A tag with POPUP set to the value used in NAME.
TITLE=text	The text to place in an annotation title. This is specified in the link using the NAME option. Display the annotation by using another A tag with POPUP set to the value used in NAME.

Parameter	Description
COLOR=color	An RGB color to use when using the POPUP or EMBED option. The default is yellow for POPUP and blue for EMBED.
NORTF	Links with an HREF tag are placed in RTF files (when using that option) by default. Use the NORTF option to prevent the link from showing in the RTF output. The text will still appear but there will be no active link in this case.
LANGUAGE=text	Text is assumed to be Unicode (2-character format) when this is used. A 2-character ISO 639 language code - for example, EN for English or JA for Japanese. The complete list of codes are available through http://www.iso.ch.
COUNTRY=text	Optional, used with the LANGUAGE option above. A 2-character ISO 3166 country code - for example, US for the United States or JP for Japan.

For example:

```
.
.
<TEXT>
<A NAME="linkhere">Here is some text I want to link back to.
</TEXT>
<PAGE>
.
.
Click <A HREF="#linkhere">here</A> to jump to the page with the anchor.
```

To insert a web link or email simply enter the destination and remember to close the tag.

For example:

```
<TEXT>
Click <A HREF="http://www.mysite.com">right here</A> to visit our site.
Click <A HREF="mailto:me@mysite.com">here</A> to send me an email.
</TEXT>
```

To insert an annotation use the NAME and ANNOT parameters to enter the text of the annotation.
Then link to the annotation by specifying the name in the POPUP parameter like this:

```
<TEXT>
<A NAME="myannot" ANNOT="Here is some text to pop-up in the document.">
Click this icon to read annotation ---> <A POPUP="myannot">.
</TEXT>
```

Click this icon to read annotation ---> .

You may also have an annotation linked to a word or phrase. The user can double-click on the link to display the popup. The link uses the LINKCOLOR for the text but is not underlined. Acrobat 4 and 5 differ slightly on how this is displayed. Acrobat 4 will paint the background (default is yellow) behind the text. Acrobat 5 does not do this. You may want to use the BGCOLOR to provide more of a visual clue to the user to double-click the link and give a consistent look between different versions of Acrobat. For example:

```
<TEXT>
<A NAME="p5" ANNOT="This could be an explanation of the highlight."
TITLE="Test Title">
Double-click on <A COLOR=lime HREF="popup:p5"><BGCOLOR VALUE=lime>this text
<BGCOLOR VALUE=white></A> to read the pop-up.
</TEXT>
```

Double-click on this text to read the pop-up.

```
<TEXT>
Double-click this tag -> <A EMBED="c:\files\myexcel.xls"
FILENAME="CorpEarnings.xls" COLOR=aqua ICON=tag MIME="application/vnd.ms-
excel" SUBJECT="Earnings for current year"> to open Excel file or right
click to open or save.
</TEXT>
```

Double-click this tag -> to open Excel file or right click to open or save.

**<ALIGN
VALUE=text>**

Used to set the current alignment. Setting the ALIGN value affects the current line so you should place a BR before a long section of text using the ALIGN tag. You may unintentionally overlap text on the same line otherwise. You may not want to use BR tags prior for headings or other circumstances where you want some left and some right aligned text on the same line.

Parameter	Description
ALIGN=L|R|C|J	Sets the alignment to Left, Center, Right or Justified.

For example:

```
<TEXT X=15 X2=65>
Text on the left
<ALIGN VALUE=C>
Centered Text
<ALIGN VALUE=R>
Text on the right<BR>
</TEXT>
```

Text on the left   Centered Text   Text on the right

<BASELINE>

Used to set the baseline in table cells when using VALIGN="M" (vertical align middle). Normally, when text is vertically aligned to the middle, the mid-point of the block of text is placed at the middle of the table cell. However, you might want to set the mid-point manually in some cases. For example, you might have a variable sized image with caption text underneath in a table cell. You can use the BASELINE tag to align the base of the images in a table row like this:

 Caption Line 1	 Caption Line 1 Line 2 Last Line	 Caption Line 1 Line 2

Note how the images are aligned at the bottom and the text is aligned at the top within the same cell. You may place the BASELINE tag anywhere in the text but only one BASELINE tag may be in each cell. Be sure to use the VALIGN=M option on the table or row. Here is how the table cells are setup for the above example:

```
<TABLE BORDER=1 VALIGN=M ALIGN=C>
<TR><TD>
<IMG SRC="smallstr.jpg">
<BR>
<BASELINE>
Caption Line 1
</TD><TD>
<IMG SRC="smallstr.jpg" SCALE=300>
<BR>
<BASELINE>
Caption Line 1
<BR>Line 2<BR>Last Line
</TD><TD>
<IMG SRC="smallstr.jpg" SCALE=200>
<BR>
<BASELINE>
Caption Line 1<BR>Line 2
</TD></TR>
</TABLE>
```

<B>
</B>

Used to turn bold face font on and off. You may also use a <STRONG> tag in place of this.

To use this feature with fonts you've added (see the ADDFONT tag), you may either add a total of 4 fonts in the following order:
Regular (i.e. myfont-regular.ttf)
Bold (i.e. myfont-bold.ttf)
Italic (i.e. myfont-italic.ttf)
Bold-Italic (i.e. myfont-bolditalic.ttf)

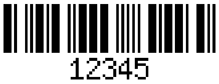
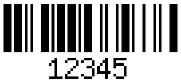
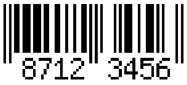
Or, add the regular version of the font and specify the option AUTOBI (for Auto Bold-Italics) as an option on the ADDFONT tag. PDF Report Writer will then be able to select the appropriate font for bold text.

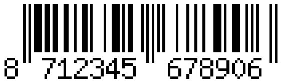
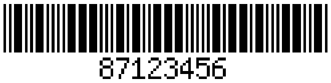
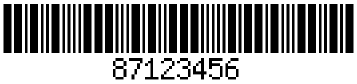
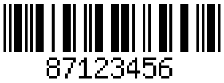
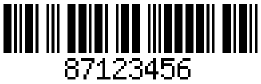
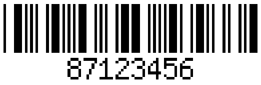
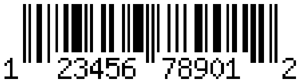
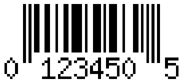
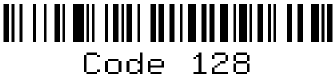
Here is some <B>bold</B> text.
Here is some **bold** text.

```
<BARCODE
    TYPE=text
    FACE=text
    HEIGHT=number
    BARWIDTH=number
    SIZE=number
    SCALE=number
    SCALEX=number
    SCALEY=number
    ROTATE=number
    VALUE=text
    SUPP=text
    NOTEXT
    ECC=text
    VERSION=number
    MODSIZE=number>
```

This tag converts the value supplied into the correct character string for the barcode. There are two categories of barcodes that PDF Report Writer can generate. The first is a text barcode that uses a TrueType font to render the barcode characters. The second is a graphical barcode which generates an image to place in the PDF as the barcode.

If you do not include the HEIGHT option, you must use UPCA as the barcode type. This implies you are using the text method or TrueType font for the barcode characters. The following types are available if you do include the HEIGHT option. In this case you are using an image for the barcode:

Type	Sample
Code39	<BARCODE TYPE='Code39' VALUE='12345' HEIGHT=30> 
COOP2of5	<BARCODE TYPE='COOP2of5' VALUE='12345' HEIGHT=30> 
EAN8	<BARCODE TYPE='EAN8' VALUE='87123456' HEIGHT=30> 

Type	Sample
EAN13	<BARCODE TYPE='EAN13' VALUE='8712345678906' HEIGHT=30> 
IATA2of5	<BARCODE TYPE='IATA2of5' VALUE='87123456' HEIGHT=30> 
Industrial2of5	<BARCODE TYPE='Industrial2of5' VALUE='87123456' HEIGHT=30> 
ITF	<BARCODE TYPE='ITF' VALUE='87123456' HEIGHT=30> 
Matrix2of5	<BARCODE TYPE='Matrix2of5' VALUE='87123456' HEIGHT=30> 
NW7	<BARCODE TYPE='NW7' VALUE='87123456' HEIGHT=30> 
UPCA	<BARCODE TYPE='UPCA' VALUE='123456789012' HEIGHT=30> 
UPCE	<BARCODE TYPE='UPCE' VALUE='123450' HEIGHT=30> 
128	<BARCODE TYPE='128' VALUE='Code 128' HEIGHT=30> 

Type	Sample
QRCode	<BARCODE TYPE='QRCode' VALUE='https://www.fytek.com' SCALE=150>

Parameter	Description
TYPE=text	Must be set to value UPCA when HEIGHT is not specified. Otherwise, may set to one of the following: COOP2of5 EAN8 EAN13 IATA2of5 Industrial2of5 ITF Matrix2of5 NW7 UPCA UPCE 128 QRCode
FACE=text	The font number or ID of the TrueType, OpenType or Type1 added font. See the ADDFONT tag for adding fonts. Use the ADDFONT tag to add the barcode font for use with this tag.
HEIGHT=number	Pixel size (at 72-DPI) for the barcode. Use SCALE (or SCALEX/SCALEY) instead for QRCode barcodes.
BARWIDTH=number	Only for Code 128 barcodes. Specify the width of the smallest bar. Default is 1.
SCALE=number	The amount to compress or expand the barcode image by in the X and Y direction. Values less than 100 will compress and values greater than 100 will expand. Setting this value overrides any values specified for SCALEX or SCALEY.

Parameter	Description
SCALEX=number	The amount to compress or expand the barcode image by in the X direction. Values less that 100 will compress and values greater than 100 will expand.
SCALEY=number	The amount to compress or expand the barcode image by in the Y direction. Values less that 100 will compress and values greater than 100 will expand.
SIZE=number	Font point size to use for the barcode.
ROTATE=number	Enter the number of degrees (0 - 360) to rotate the barcode. This is only for use with graphic barcodes (where HEIGHT is specified in the BARCODE tag).
VALUE=text	The barcode value. You may enter 11 or 12 digits for the UPCA barcode. The check digit will be automatically calculated if only 11 digits are entered.
SUPP=text	The optional 2 or 5 character supplemental value for UPCA barcodes.
NOTEXT	Set this to specify you do not want text included with the barcode. This is only for use with graphic barcodes (where HEIGHT is specified in the BARCODE tag).

<BGCOLOR
VALUE=color
SHADING=text>

Used to set the background color for highlighting text.

Parameter	Description
VALUE=color	Enter a valid color.
SHADING=text	A shading pattern to use. Set to the NAME value from the SHADING tag.

Here is some text that has been highlighted using the BGCOLOR command.

<BIG>
</BIG>

Used to increase the current point size by 2 points. You may nest these tags. Use the /BIG tag to bring the current point size down by 2 points.

This text is getting bigger

<BR
VALUE=number>

Used to insert a line break. You may optionally specify a value which is the number of inches (or centimeters if METRIC is used) to drop down from the current position.

Parameter	Description
VALUE=number	Number of inches (or centimeters if METRIC is used) to drop down. May use a decimal value.

<BULLET
VALUE=text>

Used to display a bullet point. You may also use LI (List Item) in place of BULLET.

Parameter	Description
VALUE=text	The type of bullet - Round is the default.

Sample	Type
•	Round
■	Square
●	Large
★	Star
◆	Diamond
☛	Hand1
☞	Hand2
❄	Snow
✿	Flower1
❁	Flower2
➔	Arrow1
→	Arrow2
➞	Arrow3
➢	Arrow4
➩	Arrow5
➽	Arrow6

<CAPTUREY>
NAME=text
VALUE=number
IFLESS
IFGREATER
CLEARALL>

Used to capture the current Y value. The RESTOREY command uses the last value captured with this command.

Parameter	Description
NAME=text	Optional name to use to store more than one instance. Default is blank.
VALUE=number	Used to force a value in for Y. Typically you'd leave this option off. This is mainly so you can set a min or max value for use with the MIN/MAX options in RESTOREY.
IFLESS	Only store the Y value if it's less than the currently stored value for the NAME provided.
IFGREATER	Only store the Y value if it's greater than the currently stored value for the NAME provided.
CLEARALL	Clears out (sets to 0) all the currently stored NAME values for Y positions.

<CENTER>
</CENTER>

Used to set centering on or off. When set to on, text defaults to centered and tables default to center alignment. Just the table itself - not the contents of the table cells.

<CHECK
VALUE=number>

Used to display a check mark.

Parameter	Description
VALUE=number	The type of check to print. There are six types numbered from 1 to 6.

Sample	Type
✓	1 (Default)
✔	2
✕	3
✖	4
✗	5
✘	6

<COMP
VALUE=number>

Used to specify the compression percentage for text. Unlike the other tags in this section, this tag goes outside of the TEXT block. You may override the value on the TEXT command.

Parameter	Description
VALUE=number	A percentage to compress the text by. A value less than 100 compresses text while a value greater than 100 expands text.

This text has been expanded by 150 percent of it's original size. Text will remain at this percentage until another COMP tag is issued.

<DATE
FORMAT=text>

Use to set the format for the date variable. This command just specifies how you want the date to be displayed. It does not actually display the date. You must issue it before using the &date variable however.

Parameter	Description	
FORMAT=text	A quoted text string formatted how you want the date to appear. Leaving this out defaults to Mon D, YYYY. You can use any of the following as variables:	
	Month	the full month (i.e. January)
	Mon	first 3 characters of the month (i.e. Jan)
	MM	the month number with a leading zero when needed
	M	the month number without a leading zero
	DD	the day with a leading zero when needed
	D	the day without a leading zero
	YYYY	the full four digit year
	YY	the last two digits of the year

Examples:

Format	Result
<DATE FORMAT="MM/DD/YY">	06/11/23
<DATE FORMAT="DD/MM/YY">	11/06/23
<DATE FORMAT="M-D-YYYY">	6-11-2023
<DATE FORMAT="Month, YYYY">	June, 2023
<DATE FORMAT="MM.DD.YY">	06.11.23
<DATE>	Jun 11, 2023

<EXT
HREF=text>
</EXT>

Used to add a link to an external file or application. Specify a document (like a Word, Excel, CAD, movie or other file) to be opened when the link is clicked. The file is not embedded in the PDF.

See the A tag and the EMBED option if you want to embed the external file in the PDF. That way, the file is included as part of the PDF for the end user to open or save separately if they choose.

Parameter	Description
HREF=text	Name of an external file or application to launch.

<FCOLOR
VALUE=color>

Used to change the text color.

Parameter	Description
VALUE	Enter a valid color.

Here is a text line using several different colors.

<FONT
FACE=text
SIZE=number
COLOR=color
COMP=number
CURRLINE>

Used to set the current font and/or point size. Both parameters are optional however you must at least enter either FACE or SIZE. You may nest levels of FONT and use </FONT> to return to the previous setting.

Parameter	Description
FACE=text	The font name or number.
SIZE=number	The point size for the font.
COLOR=color	Enter a valid color.
COMP=number	A percentage to compress the text by. A value less than 100 compresses text while a value greater than 100 expands text.
CURRLINE	The font change should apply only to the current line. For example, in a news article where the first line of text is a larger sized font than the rest of the article.

There are 14 built-in fonts (plus 8 Japanese, 12 Chinese and 8 Korean fonts) and you may add your own TrueType, OpenType or Type 1 fonts. Here are the built-in fonts and their values. Note that for fonts 1-14 you may specify either the number or the name for the FACE value. You may use your own TrueType fonts with languages such as Chinese as well. In this case, use the UNICODE and SUBSET options on the ADDFONT tag to keep the overall size of the PDF down. Optionally use the AUTOBI option on the ADDFONT tag to create any needed bold or italics text.

Number	Name	Sample
1	Courier	ABCDEFG abcdefg 12345
2	Helvetica	ABCDEFG abcdefg 12345
3	Times	ABCDEFG abcdefg 12345
4	Courier-Bold	**ABCDEFG abcdefg 12345**
5	Helvetica-Bold	**ABCDEFG abcdefg 12345**

Number	Name	Sample
6	Times-Bold	**ABCDEFG abcdefg 12345**
7	Courier-Italic	*ABCDEFG abcdefg 12345*
8	Helvetica-Italic	*ABCDEFG abcdefg 12345*
9	Times-Italic	*ABCDEFG abcdefg 12345*
10	Courier-Bold-Italic	***ABCDEFG abcdefg 12345***
11	Helvetica-Bold-Italic	***ABCDEFG abcdefg 12345***
12	Times-Bold-Italics	***ABCDEFG abcdefg 12345***
13	Symbol	ΑΒΧΔΕΦΓ αβχδεφγ 12345
14	ZapfDingbats	✡✢✣✤✥✦✧ ❁❂❃❄❅❆❇ ✑✒✓✔✕

The following Asian fonts are also available. You'll need to install the Chinese, Japanese or Korean font packs from Adobe in order to view a PDF with these characters. The font packs are available (free of charge) at: http://www.adobe.com/products/acrobat/acrrasianfontpack.html

Only use these fonts if your input has been converted to the encodings shown. Do not use for UTF-8 encoded input. If your input is in UTF-8 format and you have a META tag to specify that or are using -utf8 (command line) or UTF8 (method) then you do not need to specify these font names. The correct font will be used in this case. See the sample file "sample_utf8.frw" included with the download.

Alternatively, you may use a font such as arialuni.ttf and specify the UNICODE and SUBSET options on the ADDFONT tag to subset the font. By using subsetting, your glyphs will be included in the PDF and users of devices such as an iPad will be able to see the correct characters. This method is recommended if you want your PDF to be viewable on devices other than a desktop computer.

C1	STSong-Light (Chinese font) GBK-EUC-H encoding	

Number	Name	Sample
C1B	STSong-Light Bold (Chinese font)	
C1I	STSong-Light Italics (Chinese font)	
C1BI	STSong-Light Bold-Italics (Chinese font)	
C2	MSung-Light (Chinese font) ETen-B5-H encoding	
C2B	MSung-Light Bold (Chinese font)	
C2I	MSung-Light Italics (Chinese font)	
C2BI	MSung-Light Bold-Italics (Chinese font)	
C3	MSung-Light (Chinese font) ETen-B5-H encoding	
C3B	MHei-Medium Bold (Chinese font)	
C3I	MHei-Medium Italics (Chinese font)	
C3BI	MHei-Medium Bold-Italics (Chinese font)	
J1	HeiseiMin-W3 (Japanese font) 90ms-RKSJ-H encoding	
J1B	HeiseiMin-W3 Bold (Japanese font)	
J1I	HeiseiMin-W3 Italics (Japanese font)	
J1BI	HeiseiMin-W3 Bold-Italics (Japanese font)	
J2	HeiseiKakuGo-W5 (Japanese font) 90ms-RKSJ-H encoding	
J2B	HeiseiKakuGo-W5 Bold (Japanese font)	
J2I	HeiseiKakuGo-W5 Italics (Japanese font)	
J2BI	HeiseiKakuGo-W5 Bold-Italics (Japanese font)	
K1	HYGoThic-Medium (Korean font) KSC-EUC-H encoding	
K1B	HYGoThic-Medium Bold (Korean font)	
K1I	HYGoThic-Medium Italics (Korean font)	
K1BI	HYGoThic-Medium Bold-Italics (Korean font)	
K2	HYSMyeongJo-Medium (Korean font) KSC-EUC-H encoding	
K2B	HYSMyeongJo-Medium Bold (Korean font)	
K2I	HYSMyeongJo-Medium Italics (Korean font)	

Number	Name	Sample
K2BI	HYSMyeongJo-Medium Bold-Italics (Korean font)	

<I>
</I>

Used to turn italics on and off. You may also use a <EM> tag (emphesis) in place of this.

To use this feature with fonts you've added (see the ADDFONT tag), you may either add a total of 4 fonts in the following order:
Regular (i.e. myfont-regular.ttf)
Bold (i.e. myfont-bold.ttf)
Italic (i.e. myfont-italic.ttf)
Bold-Italic (i.e. myfont-bolditalic.ttf)

Or, add the regular version of the font and specify the option AUTOBI (for Auto Bold-Italics) as an option on the ADDFONT tag. PDF Report Writer will then be able to select the appropriate font for italic text.

Here is some <I>italic</I> text.
Here is some *italic* text.

```
<IMG
    SRC="text"
    ALIGN=text
    X=number
    Y=number
    X2=number
    Y2=number
    SCALE=number
    SCALEX=number
    SCALEY=number
    MAXSCALE=number
    MAXSCALEX=number
    MAXSCALEY=number
    NOTEXTSCALE
    WIDTH=number
    HEIGHT=number
    PCTX=number
    PCTY=number
    SMASK
    CACHE
    3DSRC=text
    3DBG=color
    3DZPOS=number
    3DMATRIX=text
    3DXOFF=number
    3DYOFF=number
    3DZOFF=number
    3DFOV=number
    3DNAME=text
    3DZPOSn=number
    3DMATRIXn=text
    3DXOFFn=number
    3DYOFFn=number
    3DZOFFn=number
    3DFOVn=number
    3DNAMEn=text
    3DJS=text
    3DA=text
    BORDER=number
    BORDERCOLOR=color
    PORTRAIT=text
    LANDSCAPE=text
    KEEPRATIO
    ALLPAGES|PAGE=number
    INTERPOLATE
    PAGEBREAK
    HMARGIN=number
```

VMARGIN=number
TRANSPARENCY=number[,number]
BACKGROUND
NOCONV
ROTATE=degrees
DPI=number
PERL=text>

Used to insert an image. This can be a jpeg, gif, bmp, png, etc. You may issue this command from within a text block or outside of it. Images are stored in the PDF at the same resolution (DPI) as provided.

The options starting with 3D are for 3D artwork. These files must be in U3D format. The only required option for 3D files is 3DSRC.

Parameter	Description
SRC="text"	Only required option for the tag. The path and file name of the image you wish to include. The image itself will be embedded in the PDF. Place quotes around this value. You may also use a web address that starts with http:// to pull an image from a web site. Note that the image will be stored locally while the PDF is built. Also, downloading large images over a slow modem will take time.
ALIGN=text	Only applies in a block of text. You may set this value to Bottom (default), Middle or Top. For Bottom, the bottom of the image is aligned with the bottom of the current text line. For Middle, the middle of the image is aligned with the middle of the current text line. For Top, the top of the image is aligned with the top of the current text line.
X=number	The X grid position of the left side to place the image. If you are in a block of text you don't need to specify this. The image will appear in the line of text.

Parameter	Description
Y=number	The Y grid position for the bottom of the image. If you are in a block of text you don't need to specify this. The image will appear in the line of text.
X2=number	Optional. Only for images not inside of a block of text or table cell. The X2 grid position of the right side to place the image. The image will be stretched or compressed width wise to fit between the X and X2 values. Be sure to specify the Y2 setting as well. This option overrides the scale settings.
Y2=number	Optional. Only for images not inside of a block of text or table cell. The Y2 grid position for the top of the image. The image will be stretched or compressed height wise to fit between the Y and Y2 values. Be sure to specify the X2 setting as well. This option overrides the scale settings.
SCALE=number	The amount to compress or expand the image by in the X and Y direction. Values less than 100 will compress and values greater than 100 will expand. Setting this value overrides any values specified for SCALEX or SCALEY.
SCALEX=number	The amount to compress or expand the image by in the X direction. Values less that 100 will compress and values greater than 100 will expand.
SCALEY=number	The amount to compress or expand the image by in the Y direction. Values less that 100 will compress and values greater than 100 will expand.

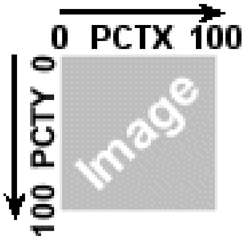
Parameter	Description
MAXSCALE=number	The maximum amount to expand the image by in the X and Y direction. Use with HEIGHT and WIDTH when you don't want the image to be scaled bigger than this amount. Use 100 for 100%.
MAXSCALEX=number	The maximum amount to expand the image by in the X direction. Use with WIDTH when you don't want the image to be scaled bigger than this amount along the X axis. Use 100 for 100%.
MAXSCALEY=number	The maximum amount to expand the image by in the Y direction. Use with HEIGHT when you don't want the image to be scaled bigger than this amount along the Y axis. Use 100 for 100%.
NOTEXTSCALE	Text in tables may sometimes need to be compressed in order to fit the cell width. This option prevents the image from being scaled along with the text.
WIDTH=number	Optional. The width to display the image at in points (1/72 of an inch). Overrides the SCALE settings. May want to set the KEEPRATIO option when setting this value.
HEIGHT=number	Optional. The height to display the image at in points (1/72 of an inch). Overrides the SCALE settings. May want to set the KEEPRATIO option when setting this value.

Parameter	Description

PCTX=number
PCTY=number
	Use these options to place an image at a particular X,Y position on the page without regards to the current GRID settings. The X and Y values in this case will represent inches (or centimeters if METRIC is used) from the top left corner of the physical page when using PCTX and/or PCTY rather than grid positions. The PCTX and PCTY values represent a percentage, 0 to 100, of the total width and height respectively from the top left corner of the image. The point determined by PCTX and PCTY is the point on the image which will be placed at position X,Y. The PCTX and PCTY options only work when the IMG tag is outside of any TEXT block or TD cell.

SMASK	Specify this option to use the included soft-mask (transparency) for PNG files. This is a secondary image used to define the transparency for the PNG. Not all PNGs contain this information so only use this option if the transparency is stored with the PNG and you require the transparency. Note that using this option alters the graphics state for the page and tends to cause normal text on the page to render darker than usual.

Parameter	Description
CACHE	Use this option on PNG, TIF, or GIF images that do not change often but are often included in your PDF files. This will cut down on the time needed to convert the image to PDF format. For example, you might use this for logos or similar images that are static and included in multiple runs of the program. A converted image file is stored in the temporary directory with an extention of .rw$. The cached version is automatically re-created if the size or date/time stamp changes on the base image. The CACHEALLIMG tag can be used to specify all images in the PDF are to be treated as if the CACHE option is set.



For example, use X=8.5 Y=5.5 PCTX=100 PCTY=50 To place an image vertically in the middle at the right edge of a page which is 8.5 inches by 11 inches. With the above page size, X=0 Y=0 PCTX=0 PCTY=0 will place an image in the upper left corner. X=8.5 Y=11 PCTX=100 PCTY=100 will place the image in the lower right corner.

Parameter	Description
3DSRC=text	Used to place a 3D CAD file in the PDF. The SRC option for the IMG tag in this case is used as the default picture for the CAD file. The file must be in U3D (Universal 3D) format. This format was developed by the 3D Industry Forum (http://www.3dif.org). This is the only 3D option that is required. See the example in this PDF.
3DBG=color	Optional. Background color for the 3D file.
3DZPOS=number	Optional. A non-negative number indicating a distance in the camera coordinate system along the z axis to the center of orbit for this view. This is the distance from the camera to the center of orbit for the 3D view, which is the point around which the camera should rotate when performing an orbit-style navigation. Default is 300.

Parameter	Description
3DMATRIX=text	A set of 12 numbers, space separated used for the camera coordinate system. You may leave this out and the program will use a default matrix of "1 0 0 0 0 1 0 -1 0 0 300 0". Or, set to a single number from 1 to 6 for other default matrices. Positions are defined in terms of pairs of x and y coordinates on the Cartesian plane for two-dimensional graphics. The origin of the plane specifies the location (0, 0); x values increase to the right and y values increase upward. For three-dimensional graphics, a third axis, the z axis, is required. The origin is therefore at (0, 0, 0); positive z values increase going into the page. The values are expressed as *a b c d e f g h i tx ty tz*. They are used as follows for the transformation. $x' = a*x+d*y+g*z+tx$ $y' = b*x+e*y+h*z+ty$ $z' = c*x+f*y+i*z+tz$
3DXOFF=number	Optional. An offset (positive or negative) for the *tx* matrix value when using a default matrix.
3DYOFF=number	Optional. An offset (positive or negative) for the *ty* matrix value when using a default matrix.
3DZOFF=number	Optional. An offset (positive or negative) for the *tz* matrix value when using a default matrix.

Parameter	Description
3DFOV=number	Optional. A number between 0 and 180, inclusive, specifying the field of view of the virtual camera, in degrees. It defines a cone in 3D space centered around the z axis and a circle where the cone intersects the near clipping plane.
3DNAME=text	Optional. This is a text string for the label the user will see for the default view of the 3D object.
3DZPOSn=text	Optional. The n in this case is a number, starting from 1 for alternate matrices. For example, 3DZPOS1="..." 3DZPOS2="..." and so on. Like 3DZPOS, this is a non-negative number indicating a distance in the camera coordinate system along the z axis to the center of orbit for this view. Default is 300.
3DMATRIXn=text	Optional. The n in this case is a number, starting from 1 for alternate matrices. For example, 3DMATRIX1="..." 3DMATRIX2="..." and so on. Like 3DMATRIX, this is a set of 12 numbers, space separated used for the camera coordinate system. Used to create alternate views of the 3D object.
3DXOFFn=number	Optional. The n in this case is a number, starting from 1 for alternate matrices. For example, 3DXOFF1="..." 3DXOFF2="..." and so on. An offset (positive or negative) for the *tx* matrix value when using a default matrix.

Parameter	Description
3DYOFFn=number	Optional. The n in this case is a number, starting from 1 for alternate matrices. For example, 3DYOFF1="..." 3DYOFF2="..." and so on. An offset (positive or negative) for the *ty* matrix value when using a default matrix.
3DZOFFn=number	Optional. The n in this case is a number, starting from 1 for alternate matrices. For example, 3DZOFF1="..." 3DZOFF2="..." and so on. An offset (positive or negative) for the *tz* matrix value when using a default matrix.
3DFOVn=number	Optional. The n in this case is a number, starting from 1 for alternate matrices. For example, 3DFOV1="..." 3DFOV2="..." and so on. Like 3DFOV, a number between 0 and 180, inclusive, specifying the field of view of the virtual camera, in degrees. It defines a cone in 3D space centered around the z axis and a circle where the cone intersects the near clipping plane.
3DNAMEn=text	Optional. The n in this case is a number, starting from 1 for alternate matrices. For example, 3DNAME1="..." 3DNAME2="..." and so on. This is a text string for the label the user will see for the particular matrix or view.
3DJS=text	Optional. Path and file name for a JavaScript script that should be executed when the 3D stream is read to create an instance of the artwork.

Text Commands

Parameter	Description
3DA=text	Optional. The circumstances under which the 3D annotation should be acivated. Valid values are: PO The annotation should be activated as soon as the page containing the annotation is opened. PV The annotation should be activated as soon as any part of the page containing the annotation becomes visible. XA The annotation should remain inactive until explicitly activated by a script or user action (this is the default).
BORDER=number	Adds a border to image using the specified thickness. Set the value of BORDER to the thickness of the border.
BORDERCOLOR=color	Specify the border color by name (black, white, blue, ...) or in HEX format (#99CC33, #7B68EE, ...).
PORTRAIT=text	Force the image to portrait mode where the width is smaller than the height. The image will be rotated 90 degrees to make it portrait if it not already portrait. Pass in CW to rotate landscape images clockwise or CCW to rotate counter clockwise. You may wish to also set the KEEPRATIO option with this option.

Parameter	Description
LANDSCAPE=text	Force the image to landscape mode where the height is smaller than the width. The image will be rotated 90 degrees to make it landscape if it not already landscape. Pass in CW to rotate portrait images clockwise or CCW to rotate counter clockwise. You may wish to also set the KEEPRATIO option with this option.
KEEPRATIO	Preserves the aspect ratio when scaling an image.
ALLPAGES|PAGE=number	ALLPAGES breaks out all of the images from a multi-page TIF. Has the same effect as putting one image tag after another with each tag specifying one page of the TIF. Images in a table cell with the ALLPAGES option will be compressed so all images fit on the page. Use a TEXT block rather than a table to spread the images across pages (when not all images will fit on a page). PAGE=number will display only the specified page from the TIF.

Parameter	Description
INTERPOLATE	Applies an image interpolation algorithm during rendering. This helps to smooth out the transition between pixels when the resolution of the image is lower than that of the output device. At higher magnification levels images on the screen typically appear jagged. This option removes most of the jagged appearance. Here's an example (second image uses INTERPOLATE):
PAGEBREAK	Page breaks between each image. Used for multi-page TIF images along with the ALLPAGES option. Must be in a TEXT block. This option is ignored if the image tag appears in a table cell.
HMARGIN=number	Used when spacing text around the image. Set a value based on the current GRID setting. Text will remain this distance from the left or right edge of the image. See the example.
VMARGIN=number	Used when spacing text around the image. Set a value based on the current GRID setting. Text will remain this distance from the bottom of the image. See the example.

Parameter	Description
TRANSPARENCY=number [,number]	Used to specify the transparency for GIF images. The transparency value, if any, is taken from the GIF file if this option is not included. This option overrides the value from the file. Specify the palette entry and, optionally, a thru value to make the entire range transparent.
BACKGROUND	Used to place the image behind any background created from a PDFPAGE or USEPAGE command. Otherwise, using this option places the image behind all other standard page content. This option is only for IMG tags outside of any text or table structures.
BLUR	Special effect to make image appear blurred.
NOCONV	Specify this option only when your image is not working properly by default. This option will bypass some of the internal conversions which are usually necessary but may cause problems with certain high-resolution images.
ROTATE=number	Enter the number of degrees (0 - 360) to rotate the image.

Parameter	Description
DPI=number	Assigns the DPI (Dots Per Inch) for the image. Must be a value greater than 72. Report Writer will typically read this value from the image. Use this option only if you want to override the computed value. The value is only used for scaling purposes - setting this value does not physically change the DPI of the image. Image heights and widths are scaled in the program by a factor of 72/DPI even when this option isn't used. For example, if your image is 600 DPI, it is scaled so when you zoom in on the PDF to 800% you see the image at approximately normal scale (100%).
PERL=text	Pass in a Perl statement or sub program to modify the image height or width. The following variables are available to the Perl code: $h - the image height $w - the image width $r - the image rotation (reference only) You may reference any of the values above using the name shown. Their default values are from any computations made just prior to rendering the image. Only $h and $w are changable. For example, if you modify $h = 100 and $w = 200 then the image will be scaled to a height of 100 and width of 200. The -allowperl option or setAllowPerl method must be specified when using this option.

Here is an example of using an image with a block of text. The text will automatically space over to leave room for the image when the HMARGIN and/or VMARGIN tags are used with the image. The image may only be placed at the left or right when using this option. Set the alignment to the right if you want the image right aligned. Right after the image, use the Y tag along with an ALIGN option for the text that follows.

Here is some text to place next to the image to demonstrate the HMARGIN and VMARGIN options. Note that the text will be moved to the right until it is passed the image. The text will then flow into its normal position after it has moved past the image. You must place a <Y ALIGN=text> tag after the IMG tag in order for this to work. In addition, the IMG tag must have the HMARGIN and/or VMARGIN set.

Here is some text to place next to the image, this time with the image on the right. Note that the text will be moved to the right until it is passed the image. The text will then flow into its normal position after it has moved past the image. You must place a <Y ALIGN=text> tag after the IMG tag in order for this to work. In addition, the IMG tag must have the HMARGIN and/or VMARGIN set.

The above is coded as:

```
<TEXT FACE=15 SIZE=11 ALIGN=J X=15 X2=65 COMP=100><BR>
<IMG SRC="star.tif" HMARGIN=2 VMARGIN=2><Y ALIGN=J>
Here is some text to place next to the image to demonstrate the HMARGIN and VMARGIN options.
Note that the text will be moved to the right until it is passed the image.
The text will then flow into its normal position after it has moved past the image.
You must place a &lt;Y ALIGN=text&gt; tag after the IMG tag in order for this to work.
In addition, the IMG tag must have the HMARGIN and/or VMARGIN set.<BR><BR>
<ALIGN VALUE=R><IMG SRC="star.tif" HMARGIN=2 VMARGIN=2><Y ALIGN=J>
Here is some text to place next to the image, this time with the image on the right.
Note that the text will be moved to the right until it is passed the image.
The text will then flow into its normal position after it has moved past the image.
You must place a &lt;Y ALIGN=text&gt; tag after the IMG tag in order for this to work.
In addition, the IMG tag must have the HMARGIN and/or VMARGIN set.
</TEXT>
```

<LINESPACE
VALUE=number>

Used to specify the line spacing in 1/72 of an inch. You may override the value on the TEXT command.

Parameter	Description
VALUE=number	The amount of space between lines of text specified in units of 1/72 of an inch. Default is 2.

This text has a linespace value set at 12 which is one more than the current point size of 11. Notice how far each line drops down when the text wraps. This is roughly double spaced text.

<LINKCOLOR
VALUE=color>

Used to set the text and line color for links. Default is an RGB value of .2,.2,1.

Parameter	Description
VALUE=color	Sets the link color.

Here is a link with LINKCOLOR set to red: FyTek, Inc.

<LINKLINE
VALUE=number>

Used to set the line width for links. Set to 0 to turn-off underlines in links altogether. Default is 1.

Parameter	Description
VALUE=number	Line width or 0 to turn-off link underlines.

Here is a link with LINKLINE set to 3: FyTek, Inc.

Here is a link with LINKLINE set to 0: FyTek, Inc.

```
<MEDIA
    HREF=text
    X=number
    Y=number
    HEIGHT=number
    WIDTH=number
    MIME=text
    FILENAME=text
    REPEAT[=number]
    BORDER=color
    PAGEOPEN
    PAGEVISIBLE
    MOUSEOVER
    CONTROLS=text
    ALTTEXT=text>
</MEDIA>
```

Used to embed an audio or video file in the PDF. Requires Acrobat or Reader 6.0 or higher to view the multimedia file. Works similar to the A tag. Place text or an image between the opening and closing MEDIA tags. You may issue this command from within a text block or outside of it. You do not need to use the closing /MEDIA tag when outside of a block of text or table cell. The X/Y and HEIGHT/WIDTH values will need to be specified when outside of a block of text.

Place an image or text between the opening/closing tags. Clicking the image or text starts playing the clip unless one of the page open/visible or mouseover options are used.

You may leave the closing /MEDIA tag off and specify a WIDTH and HEIGHT instead. The WIDTH and HEIGHT are based on the current grid units.

Parameter	Description
HREF=text	Name of an file to embed. This may be an audio or video file.
X=number	The X position in points (1/72 of an inch) from the left page edge to place the image. If you are in a block of text you don't need to specify this. The media will appear in the line of text.
Y=number	The Y position in points (1/72 of an inch) from the bottom page edge to place the image. If you are in a block of text you don't need to specify this. The media will appear in the line of text.

Parameter	Description
HEIGHT=number	Only needed if not in a block of text or not using the closing /MEDIA tag. The HEIGHT of the media in grid units when in a block of text. The HEIGHT of the media in points (1/72 of an inch) when not in a block of text.
WIDTH=number	Only needed if not in a block of text or not using the closing /MEDIA tag. The WIDTH of the media in grid units when in a block of text. The WIDTH of the media in points (1/72 of an inch) when not in a block of text.
MIME=text	The mime type for the file. This is determined for some files automatically by the file name extension. The files currently configured are .avi, .wmv, .mpeg, .mov, .swf, .wav, and .mp3. Other types must supply the value (unless it exists in the Windows system registry at build time). For example, video/x-msvideo or audio/x-midi. Windows based versions of Report Writer will attempt to locate the correct mime type from the system registry when the PDF is built.
FILENAME=text	The file name to use for the attachment. This is not the source file name on disk - it is the filename to refer to the embedded file by once it's in the PDF. The default is the name used in the HREF entry.
REPEAT[=number]	Use REPEAT by itself to continuously repeat the video or audio clip. The default is to play the clip once then stop. Set REPEAT to a value to repeat the clip that many times, such as REPEAT=3 to play 3 times.
BORDER=color	Set to a valid color. Draws a border around the area where the media is located.
PAGEOPEN	Play the file when the page is opened in the viewer.
PAGEVISIBLE	Play the file when the page is visible in the viewer.
MOUSEOVER	Play the file when the cursor or mouse pointer is on the active area of the media.
CONTROLS=text	Set to ON or OFF. Default is ON. The controls for the media file are not shown if set to OFF.

Parameter	Description
ALTTEXT=text	Alternate text to display if the media clip cannot be played.

Media examples - requires Acrobat or Reader 6.0 or higher to view:

```
<TEXT>
<MEDIA HREF="c:\my music\jazz.mp3">
Sample music file
</MEDIA>
</TEXT>
```

Sample music file

<MINLINES
VALUE=number>

Used to force a page break if the minimum number of lines specified cannot fit on the current page. Use this command to prevent orphan headings in text blocks.

Parameter	Description
VALUE	The minimum number of lines (approximate - based on the current font size) that must be able to fit on the current page. If the minimum number doesn't fit, a page break is performed. This feature can be used in TEXT blocks to prevent a heading from appearing at the bottom of a page while the text it goes with starts on the next page. In this case, set MINLINES to a value (somewhere between 3 and 10 perhaps) and place this command just before the heading.

<PAGEFOOTER>
</PAGEFOOTER>

Use this block to print a table or text at the bottom of the page. A page break will occur if there is not enough room on the current page to print the contents. Place any TABLE or TEXT tags between the opening and closing PAGEFOOTER tags.

<MOV
HREF=text>
</MOV>

Used to add a link to an external movie file. Specify a movie file to be opened when the link is clicked. Works similar to the A tag. The movie file must be available as a separate file on the machine viewing the PDF.

Use the MEDIA tag for embedding movies or audio (such as MP3 files) in the PDF.

Parameter	Description
HREF	Name of an external movie file to launch.

<P>

Used for a new paragraph. This tag is equivalent to using <BR><BR>.

**<REND
VALUE=number>**

Used to specify the text rendering mode.

Parameter	Description
VALUE=number	0 = Fill text (default) 1 = Stroke text (outline) 2 = Fill then stroke 3 = No fill or stroke (invisible)

This text is in mode 0
This text is in mode 1
This text is in mode 2

```
<RESTOREY
    NAME=text
    MIN
    MAX>
```

Used to restore the current Y value. The last Y position stored by the CAPTUREY command is used by this command to set the current Y position.

Parameter	Description
NAME=text	The name to restore (if a name was used in the CAPTUREY command).
MIN	Sets Y to the minimum value of all currently stored values from the CAPTUREY command.
MAX	Sets Y to the maximum value of all currently stored values from the CAPTUREY command.

**<ROTATE
VALUE=number>**

Used to rotate text.

Parameter	Description
VALUE=number	The number of degrees to rotate the text. Text is rotated around the lower left point of the first character on the line.

Here is some text that has been rotated 45 degrees. This text will appear slanted on the page. One use for this would be for a watermark on the back of a page. Use a light color for the text in that situation.

<SHY>

Used to insert a soft hyphen. Place the tag within a word, such as:
en<SHY>courag<SHY>ing

The soft hyphens will be converted to a - when the word doesn't quite fit at the end of a line of text but enough of it fits along with the hyphen. Any soft hyphens not used are ignored.

May also use the &shy; variable in place of this tag. Works the same way:
en&shy;courag&shy;ing

<SMALL>
</SMALL>

Used to decrease the current point size by 2 points. You may nest these tags. Use the /SMALL tag to bring the current point size up by 2 points.

This text is getting smaller

<SUB>
</SUB>

Used to turn subscripting on and off.

Here is a line using the $_{\text{subscripting}}$ tag.

<SUP
VALUE=number>
</SUP>

Used to turn superscripting on and off.

Here is a line using the $^{\text{superscripting}}$ tag.

Parameter	Description
VALUE=number	Optional. Use this to control the height of the subscript. The amount in is units of 1/72 of an inch. The default is 5 if this parameter is left out.

<TAB
VALUE=number>

Used to offset the current X position in a line of text. Affects the current line only. If you want to change the margins use the X tag instead. Setting the TAB value affects the current line so you should place a BR before a section of text using the TAB tag. You may unintentionally overlap text on the same line otherwise.

Parameter	Description
VALUE	The amount based on the current XUNITS setting from the GRID tag to offset the current X position by. May be positive or negative.

For example:

```
<TEXT X=15 X2=65>
Here is where a line normally starts.<BR>
<TAB VALUE=-5>This line is shifted to the left by 5 units.<BR>
<TAB VALUE=5>This line is shifetd to the right by 5 units.<BR>
This line is back to where text normally starts from.<BR>
</TEXT>
```

Here is where a line normally starts.
This line is shifted to the left by 5 units.
This line is shifetd to the right by 5 units.
This line is back to where text normally starts from.

```
<TEXT
    X=number
    Y=number
    X2=number
    Y2=number
    WIDTH=number
    ALIGN=L|R|C|J
    ABS
    BORDER=number
    BORDERCOLOR=color
    BORDEROUTER
    BORDERDASH=number
     [,number,...]
    PADDING=number
    PADDINGX=number
    PADDINGX2=number
    CORNERSIZE=number
    CORNERSTYLE=text
    BGCOLOR=color
    SHADING=text
    PATTERN=number
    PATTERNCOLOR=color
    3D
    SIDES=text
    COLNUM=number
    COLSPACE=number
    COLBAR=number
    NOBREAK
    FORCE
    CLIP
    FACE=font number
    SIZE=point size
    NO_ORPHAN
    NO_WIDOW
    EXCEL=text
    EXCEL_SEQ=number
    LINESPACE=number
    BREAKON="text"
    FCOLOR=color
    SCOLOR=color
    ROTATE=number
    ANGLE=number
    COMP=number
    VALUE=text>
</TEXT>
```

Used to print a block of text. All of the parameters are optional. Place your text to print between the opening and closing text blocks. You may place TABLE tags inside of the text block. Doing so has the same effect as closing the text block, printing the table, then reopening the text block.

Parameter	Description
X=number	Sets the value of the left margin in units based on the GRID command. Default is 0.
Y=number	Sets the value for the top of the text based on the GRID command. Default is the current Y value. Generally you don't need to set the Y value. Only set this value when you're trying to place the text in a given spot, like when trying to match a pre-printed form. Text will simply start at the current position and, upon a page break, will continue on the next page at the top of the report body.
X2=number	Sets the value of the right margin in units based on the GRID command. Default is value of XUNITS from the GRID command.
Y2=number	Sets the value for the bottom of the text based on the GRID command. Generally you don't need to set the Y2 value. Only set this value when you're trying force some text to fit into a box on the page and you don't want the text to run outside of that area. Any text that doesn't fit is not printed.
WIDTH=number	Set to a percentage such as 80%. This has the effect of setting the X2 value based on a percentage of the right margin.

Parameter	Description
ABS	Normally, when using center or right alignment, text is centered or right aligned within the left and right edges of a TEXT block. The left and right edges of a TEXT block span from GRID position 0 to the current XUNITS value unless overridden with the X or X2 parameters described above. The ABS (absolute positioning) option will ignore the edges and center or right align text around the point specified by the X parameter. There is no word wrapping when using the ABS option (it sets the FORCE option) since the X and X2 parameters are no longer used for the left and right edges of the TEXT block. For example, using <TEXT ALIGN=C> will center text in the middle of the page. Using <TEXT ALIGN=C X=25 ABS> will center text at GRID position 25. You could achieve the same result without ABS by using <TEXT ALIGN=C X=0 X2=50 FORCE>.
ALIGN=L|R|C|J	Sets the initial alignment to Left, Right, Center or Justify. Default is Left.
BORDER=number	Sets the border size. A value of 1 is a line with a width of 1/72 of an inch.
BORDERCOLOR=color	Sets the border color (when the 3D option is not used).
BORDEROUTER	Extends the border from the X to the X2 position regardless of how far the text extends. Use this to force the left and right border to be at the X and X2 position rather than the default of the border being up against the left and right extent of the text.
BORDERDASH=number [,number,...]	A number or comma separated list of numbers to specify the ON/OFF length of the border.
PADDING=number	Use to provide extra padding between text and top/bottom border. Specify the value in points (1/72 of an inch).

Parameter	Description
PADDINGX=number	Use to provide extra padding between text and the left border. Specify the value in points (1/72 of an inch).
PADDINGX2=number	Use to provide extra padding between text and the right border. Specify the value in points (1/72 of an inch).
CORNERSIZE=number	The size of the optional CORNERSTYLE in inches (centimeters if METRIC is used). Consider using PADDING when setting this to a relatively large value so the text remains within the confines of the border.
CORNERSTYLE=text	The optional type of corners to use for the border. The default border is a simple rectangle. Valid values are: Round Bevel Scoop Stair In addition, you may specify a different style for any given corner. Pass a comma separated string containing a code for each corner followed by a colon then the type. The codes for each corner are: TL - Top left TR - Top right BR - Bottom right BL - Bottom left For example, set to "TL:round,TR:scoop,BR:round" to give the top left and bottom right corners a round edge and the top right a scoop edge. The bottom left edge, since it wasn't specified, will have a standard right-angled corner.
BGCOLOR=color	Sets the background color of the text block.
SHADING=text	A shading pattern to use for the background. See an example here.
PATTERN=number	A pattern number to use for the background. See the RECT command and its PATTERN option for a sample of each pattern's appearance.

Parameter	Description
PATTERNCOLOR=color	Color for the pattern.
3D	Sets the border style as 3D which is lighter on top and left and darker on bottom and right. The base color used is the BGCOLOR or BORDERCOLOR if BGCOLOR is not specified.
SIDES=text	Sets what sides to draw the border on. Default is all sides. Values are L (Left), R (Right), T (Top) and B (Bottom). For example, use SIDES=T,B to specify the border should be drawn on top and bottom only.
COLNUM=number	Sets the number of columns to divide the text into. Setting to two, for example, will fill up the left half of the page then move to top right half of the page and continue. A new page is started automatically once all columns are filled.
COLSPACE=number	Sets the amount of space in units based on the GRID command between each column.
COLBAR=number	Draws a vertical bar between multiple columns of text. Set this parameter to the line width.
NOBREAK	Set this to prevent the text block from breaking across a page. You should only use this option for a small amount of text that will fit between the header and footer. If the text cannot fit between the header and footer then it will only print the portion that fits.
FORCE	Set this to prevent the text from wrapping or flowing to the next page. You should only use this option to force some text to print where it would normally try to wrap or flow to the next page.
CLIP	Clips the text based on the X/Y and X2/Y2 values. All four of these values must be set for CLIP to work. Text may be cut-off midline based on the size of the rectangle. There is no page break in this case.

Parameter	Description
FACE=font number	Sets the font.
SIZE=point size	Sets the point size for the font.
NO_ORPHAN	Helps prevent orphan lines. An orphan is the first line of a paragraph printed by itself at the bottom of the page. You may also specify this for all text blocks by using the META tag.
NO_WIDOW	Helps prevent widow lines. A widow is the last line of a paragraph printed by itself at the top of the page. You may also specify this for all text blocks by using the META tag.
EXCEL=text	Set the worksheet (or tab) name you want this text block exported to. Also used to start a new table when exporting to HTML. See the EXCEL tag for more information.
EXCEL_SEQ=number	Optional. Used to set the ordering sequence for the tab. Tabs are arranged in order from lower number to higher based on the sequence number, when supplied.
LINESPACE=number	Sets the text line spacing. Each unit is 1/72 of an inch. Default is 2.
BREAKON="text"	Set of characters a line break may occur on in addition to the space character. This overrides any global setting made with the BREAKON tag.
FCOLOR=color	Sets the fill color. This is also the font color.
SCOLOR=color	Sets the stroke color.
ROTATE=number	The number of degrees (0-360) to rotate the text. Text is rotated counter-clockwise around the lower left point of the first character on the line. A page break does not occur when using this option if the text flows off the page.

Parameter	Description
ANGLE=number	Similar to ROTATE except the text is rotated about the center. Pass in the number of degrees (0-360) to rotate the text counter-clockwise. Or use a negative number from -360 to 0 to rotate clockwise. Also, unlike ROTATE, the background color and/or border is rotated with the text. A page break does not occur when using this option if the text flows off the page.
COMP=number	A percentage to compress the text by. A value less than 100 compresses text while a value greater than 100 expands text.
VALUE=text	Shortcut used to place a small amount of text on the page. Normally, you would place text between the opening and closing text tags and not use this option. There is no need to use the closing </TEXT> command when using this option. You may not use any tags in your text string with this option and no checks will be done for page breaks. This should only be used when you have some small text strings to place at specific X/Y locations on the current page.

The following commands in this section detail the various attributes you may apply to text.

<U>
</U>

Used to turn underlining on and off. You can use <U2> and </U2> for a double underline.

Here is a line with an underline and a double underline tag in it.

These are the variables you may use in your document. To insert the text rather than the value of the variable, use a slash in front of the &. For example, \&date to display &date in your text rather than the current date.

Variable	Sample	Description
&page	232	The current page number. This may be reset by using the SETPG command.
&pagesub	232	The current page number for the sub-section. This is set via the STARTSUB option on the PAGE tag.
&runpage	232	The running page number. Always reflects the current physical page in the document.
&totpage	512	The total number of pages in the document.
&totpagesub	512	The total number of pages in the sub-section. This is set via the STARTSUB option on the PAGE tag.

Variable	Sample	Description
&getpage("ANCHOR")	162	The page number the anchor is on. Use this feature to refer to a page elsewhere in the document. Set an anchor with the NAME parameter filled in on the page you want to refer to. Use the function &getpage("anchor name") on the page where you want to print the page number of the referenced page. The value used for the getpage function is the same value you used for the NAME parameter in the anchor. Include RUNPAGE as an option on the anchor tag to return the running (&runpage) page number. The default is the page (&page) number which can be changed with the SETPG function.
&date	Jun 11, 2023	The current system date when the document was created. See the DATE tag for formatting options.
&time	6:17 pm	The current system time when the document was created.
&reg	®	Registered trademark symbol
&trad	™	Trademark symbol
&copy	©	Copyright symbol
&emdash	—	emdash
&lt	<	Less-than symbol
&gt	>	Greater-than symbol
&plusmn	±	Plus/Minus

Variable	Sample	Description
&cent	¢	Cent
&pound	£	Pound
&euro	€	Euro
&yen	¥	Yen
&deg	°	Degree
&shy	-	Soft hyphen (same as using the SHY tag)
&sup2	^{2}	Superscript 2
&sup3	^{3}	Superscript 3
&mu	µ	Mu
&glt	«	Guillemet (left)
&ggt	»	Guillemet (right)
&Oslash	Ø	O with slash
&oslash	ø	o with slash

<X
VALUE=number
X2=number>

Used to change the X and/or X2 values in a block of text. Affects all lines until another X tag is issued. You can also use this feature to do some column alignment in a block of text. Both parameters are optional however you must at least enter either VALUE or X2. Setting the X value affects the current line so you should place a BR before a section of text using the X tag. You may unintentionally overlap text on the same line otherwise.

Parameter	Description
VALUE	The left setting based on the current XUNITS setting from the GRID tag. Must be a positive value.
X2	The right setting based on the current XUNITS setting from the GRID tag. Must be a positive value.

For example:

```
<TEXT X=15 X2=65>
Here is where a line normally prints.<BR>
<X VALUE=20 X2=30>This text has been squeezed into a smaller area.<BR>
It will stay this way until another X command is issued.<BR>
</TEXT>
```

```
Here is where a line normally prints.
      This  text  has
      been  squeezed
      into  a  smaller
      area.
      It will stay this
      way         until
      another         X
      command      is
      issued.
```

VALUE=number>

Used to change the Y position in a block of text. About the only time you'll need to modify the Y position is when you want to move back to the top of the text block for positioning some text (like a report header).

Parameter	Description
VALUE	The setting based on the current YUNITS setting from the GRID tag. Must be a positive value. The top of a block of text is always 0.

For example:

```
<TEXT X=15 X2=65>
Text on the left<BR>
Second line of text on the left<BR>
<Y VALUE=0><ALIGN VALUE=R>
Text on the right<BR>
Second line of text on the right<BR>
</TEXT>
```

Text on the left   Text on the right
Second line of text on the left   Second line of text on the right

Tables

Tables are used in Report Writer to display data in a matrix form. Columns are sized based on the amount of text to show in each column. Columns remain in a uniform size across page breaks for any given table. Tables do not expand past the right margin of the page. This means that text may be compressed in order to fit the width of the table on the page. There is an option to vertically break wide tables to allow the columns that fit to appear on one page while the rest of the columns show on another.

Tables open with a <TABLE> tag and close with </TABLE> Between these tags are the rows and data for the table. Report Writer will fit as much of the table as possible on each page. The table will stop once it reaches the bottom margin then automatically continue on the next page if necessary.

The next level down from TABLE is TBODY. TBODY is optional and is used to group rows together. You may have as many TBODY tags per table as you want but they should not be nested. The <TBODY> tag starts a TBODY section and </TBODY> closes it. You can use TBODY to apply certain colors or border attributes to a set of rows rather than on each row individually.

The table rows contain a group of one or more table cells. The <TH>, <TF> and <TR> tags start a row and </TH>, </TF> and </TR> close them. The TH (Table Header) row is used to denote the row as a header for the table. Header rows are repeated at the top of the table on page breaks. The TF (Table Footer) row is used to denote the row as a footer for the table. Footer rows are repeated at the bottom of the table on page breaks. The TR (Table Row) rows contain the body of the table and are displayed once.

Note the TF row must come before the first TR row. Even though the TF is at top as far as the table commands, it will not print until the table ends or reaches the end of the page.

The lowest level is the table cell. The <TD> starts a cell and </TD> closes it. The text or images to display goes between the opening and closing TD tag.

Here's the basic structure of a table showing the optinal LEGEND and TBODY tags:

<TABLE>			
	<LEGEND>		
		...legend text...	
	</LEGEND>		
		<TH>	
			<TD>
			...header cell contents...
			</TD>
			...more columns...
			<TD>
			...header cell contents...
			</TD>
		</TH>	
		<TF>	
			<TD>
			...footer cell contents...
			</TD>
			...more columns...
			<TD>
			...footer cell contents...
			</TD>
		</TF>	
	<TBODY>		
		<TR>	
			<TD>
			...cell contents...
			</TD>
			...more columns...
			<TD>
			...cell contents...
			</TD>
		</TR>	
		...rest of the table rows...	
	</TBODY>		
		<TR>	
			<TD>
			...cell contents...
			</TD>
			...more columns...
			<TD>
			...cell contents...
			</TD>
		</TR>	
</TABLE>			

```
<TABLE or <MODTABLE
    X=number
    X2=number
    Y=number
    Y2=number
    WIDTH=number
    HEIGHT=number
    MINCOLWIDTH=number
    MINLINES=number
    MAXLINES=number
    CELLSPACING=number
    CELLXSPACING=number
    CELLYSPACING=number
    CELLPADDING=number
    CELLXPADDING=number
    CELLYPADDING=number
    CELLWIDTHMAX=number
    AUTOCOLS=number
    CELLGROUP=number
    VARIABLECOLS
    PAGEROWS=number
    YPOS=number
    CELLSPLIT
    TBLALIGN=L|R|C
    NAME=text
    EXCEL=text
    EXCEL_SEQ=number
    COMPTEXT
    DOTTEXT
    COMP=number
    BREAK=DOWN|OVER
    SOFTBREAK=number
    WHOLETABLE
    BACKGROUND=text
    CELLBACKGROUND=text
    CELLBACKGROUND2=text
    TEXTCLIP=text
    TILE
    TILE2
    TEXTCLIPTILE
    TEXTCLIPREND=number
    BORDER=number
    BORDERCOLOR=color
    BORDERDASH=number
     [,number,...]
    BORDEROPTS=text
    CORNERSIZE=number
```

CORNERSTYLE=text
SIDES=text
TABLEBORDER=number
TABLEBORDERTOP
TABLEBORDERPADDING=
number,number,number,number
TABLEBORDERCOLOR=color
TABLEBGCOLOR=color
TABLEBORDERDASH=
number[,number,...]
TABLECORNERSIZE=number
TABLECORNERSTYLE=text
TABLEPATTERN=number
TABLEPATTERNCOLOR=color
TABLESIDES=text
TABLEFILL=color
BORDERCOLS=list
TABLEBACKGROUND=text
TILEBACKGROUND
BGCOLOR=color
STRIPE=color,color
STRIPEX=number,number
SHADING=text
PATTERN=number
PATTERNCOLOR=color
3D
COLORSTARTx=color
COLORMIDx=color
COLORENDx=color
COLORBGSTARTx=color
COLORBGMIDx=color
COLORBGENDx=color
COLORBGCONTENTSx
COLORBARx=number
COLORBARROUNDx=number
COLORBARBLOCKx=number
COLORBARIMGx=text
COLORBARTOPx=number
COLORBARBCx=color
COLORBARZEROx=number
COLORMINx=number
COLORMAXx=number
COLSORT=text
COLSORTDIR=text
BREAKONCHANGE=number
MULTISORT=list
MULTISORTDESC=list

```
    MULTISORTARROWS
    MULTISORTSIZE=number
    MULTISORTUPCOLOR=text
    MULTISORTDOWNCOLOR=text
    MULTISORTUPIMG=text
    MULTISORTDOWNIMG=text
    COLNUM=number
    COLSPACE=number
    COLBAR=number
    ALIGN=L|R|C|J
    VALIGN=T|M|B
    FACE=font number
    SIZE=point size
    LINESPACE=number
    FCOLOR=color
    SCOLOR=color
    CONTENTANGLE=number
    CONTENTANGLES=number
    CONTENTBORDER=number
    CONTENTBORDERCOLOR=color
    CONTENTBORDERBGCOLOR=color
    CONTENTBORDERPADX=number
    CONTENTBORDERPADY=number
    FLUSH
    QUICK
    FORCEWIDTH
    BREAKON="text"
    OL_LEVEL=number
    OL_DESCR=text
    OL_CLOSED
    OL_SUB_LEVEL=number
    OL_SUB_CLOSED
    OL_COL
    OL_ALL
    OL_BREAK
    OL_LENGTH
    PLAIN[=text]>
</TABLE>
```

Used to define a table. All of the parameters are optional.

Some of the parameters, like COLORSTARTx, are variable. That is, the x is replaced with a number representing a column in the table.

Columns are numbered starting from 1 on the far left and incrementing by 1 for each column to the right. To apply COLORSTARTx to column 5 in a table you would use COLORSTART5="...", for example.

The AUTOCOLS and VARIABLECOLS options give you flexibility to create more fluid layouts. AUTOCOLS will automatically determine where to place table row (TR) tags based on table content. You only need to specify a single opening TR tag, followed by all of your TD tags, then a single closing /TR tag just before the closing /TABLE tag. Use NOROWBREAK to prevent a cell from breaking to the next row.

Use MODTABLE when you're inside a table and wish to switch to a different layout. You would use this when you have two tables with different column layouts but want them both inside the same TABLEBORDER and optionally sharing the same SUBHEADING rows. The MODTABLE tag goes after the closing </TR> tag in the previous table. The YPOS=number option may be added to conditinally page break if the new layout is starting near the bottom of the page. This works the same as the YPOS option on the PAGE tag. Do not place a closing </TABLE> tag before the MODTABLE tag. For example:

```
<ALIGN VALUE=L>
<TABLE TABLEBORDER=1 WIDTH=100%>
<TR SUBHEADING=1>
<TD STARTX=0 ENDX=50>My Subheading</TD>
</TR>
<TR>
<TD>Col 1</TD>
<TD>Col 2</TD>
<TD>Col 3</TD>
</TR>
<MODTABLE WIDTH=100%>
<TR>
<TD>Col 1</TD>
<TD>Col 2</TD>
<TD>Col 3</TD>
<TD>Col 4</TD>
<TD>Col 5</TD>
</TR>
</TABLE>
```

The row for "My Subheading" will repeat on page breaks regardless of which table breaks across the page. The layout starting with MODTABLE is completely independant from the column widths used for table above it. You may have more than MODTABLE tag within a single TABLE tag.

Each MODTABLE command may have a different layout in terms of column widths. This is important to keep in mind when using a TR tag with SUBHEADING specified along with MODTABLE. In this case you may need to use CLEARSUBHEADINGS or replace a subheading that does not use STARTX and ENDX in the TD cell. Otherwise, the text in the subheading row may no longer fit within the confines of the new layout. A row from a previous TABLE or MODTABLE with SUBHEADING specified is not (by default) used in computing the column widths for the

MODTABLE tag. You may use the SIZEHEADINGS tag if you want to include these rows when computing column sizes.

Another option is to use the KEEPCOLSIZE command on the TR tag that has the SUBHEADING option. This will force the use of the column sizes from the original table just for this sub heading row. You may also want to use the NAME parameter on the TABLE and/or MODTABLE tags to retain a consistent layout.

Parameter	Description
X=number	Sets the value of the left margin in units based on the GRID value for XUNITS. Default is 0. Use a % after the number to use a percentage of the current XUNITS setting. Setting X and/or X2 does not stretch a table to fit within these values. It simply sets the margins for the maximum width for the table. To stretch the table to fit a certain width, use the WIDTH parameter.
X2=number	Sets the value of the right margin in units based on the GRID value for XUNITS. Default is the value of XUNITS from the GRID command. Use a % after the number to use a percentage of the current XUNITS setting.
Y=number	Sets the value of the top of the table based on the GRID command. Default is the current Y value. Generally you don't set the Y or Y2 values. Only set these values when you're trying to leave some extra space to make the report look more like a pre-printed form you have. A table will simply start at the current position and, upon a page break, will continue on the next page at the top of the report body.
Y2=number	Sets the value of the top of the table based on the GRID command for all subsequent pages for the table. Default is 0.
WIDTH=number	Sets the minimum width of the table based on the GRID command. Only set this parameter if you want to force the table to be wider that it would normally size itself. You can also specify a percentage of the total page width (minus the current margins). For example, set WIDTH=80% to take up 80% of the available page width.

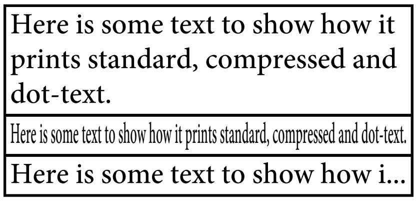
Tables

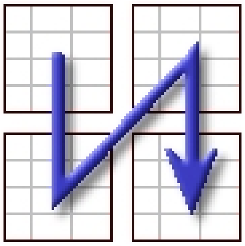
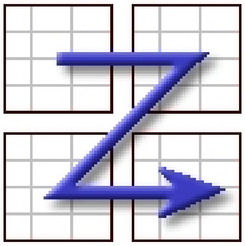
Parameter	Description
HEIGHT=number	Sets the maximum height of the table based on the GRID command. You only use this parameter when you have a table you wish to force into a particular area on the page. Tables with this parameter set will not span across pages so you should set the HEIGHT to a value less than the current YUNITS setting (from the GRID command).
MINCOLWIDTH=number	Sets the minimum width in inches (or centimeters if METRIC is used) that a column will be sized to if the table needs to be reduced to fit on the page. The default value is 1 inch. The larger this value is set to the more chance the table text will require compression.
MINLINES=number	Sets the minimum number of lines to print in each table cell. Use this only if you are trying to maintain a minimum size for each row. Setting this value to 3, for example, will force two extra blank lines to be printed in a cell that only has one line of text. No lines are added to a cell where the number of lines of text is equal to or greater than this number.
MAXLINES=number	Sets the maximum number of lines to print in each table cell. Use this only if you are trying to maintain a maximum size for each row. Setting this value to 3, for example, will truncate text that would normally print on lines 4 or after in the cell.
CELLSPACING=number	Sets the space between cells based on units of 1/72 of an inch. You may use decimal values.
CELLXSPACING=number	Sets extra horizontal space between cells based on units of 1/72 of an inch. You may use decimal values.
CELLYSPACING=number	Sets extra vertical space between cells based on units of 1/72 of an inch. You may use decimal values.
CELLPADDING=number	Sets the padding within cells based on units of 1/72 of an inch. You may use decimal values.
CELLXPADDING=number	Sets extra horizontal padding within cells based on units of 1/72 of an inch. You may use decimal values.

Parameter	Description
CELLYPADDING=number	Sets extra vertical padding within cells based on units of 1/72 of an inch. You may use decimal values.
CELLWIDTHMAX=number	Sets the maximum width of any table cell to be no more than this value based on units of 1/72 of an inch. You may also use a % after the number to mean a percentage of the table width.
AUTOCOLS=number	Use this option to pass a set of table cells (the TD tag) without specifying each row (the TR) tag other than the opening TR tag and single closing /TR tag just before the /TABLE tag. See the file sample5.frw for an example of using AUTOCOLS. Pass a number based on the current grid setting or add a % after the number to mean a percentage of the total table allocated width. If you use TH (table header) rows with this option, include either just one cell or, if using CELLGROUP, then CELLGROUP number of cells. Any additional TH cells for additional columns that fit in a single row will be created by Report Writer. For example, suppose you have a set of product thumbnails and a barcode you want to show for each image. This layout fits well in a table but you're not sure how many products will fit across on the page. By setting AUTOCOLS=95% (if your thumbnails are all roughly the same size, you might use 100%) Report Writer will fit as many thumbnails across as it can then automatically insert a TR tag. This allows you to have a more fluid layout, similar to CSS in HTML using float:left on a DIV tag. The pseudo classes first-of-type and last-of-type may be used with the CLASS option when applying a style to a cell. For example, if you have <TD CLASS='bord'> and the following style settings: <style> bord { border:1; sides:r } bord:last-of-type { border:0 } </style> This will apply a border to right of all table cells except for the last cell in each row.

Parameter	Description
CELLGROUP=number	Optionally use this option with AUTOCOLS. Groups a related set of cells (as specified by the number passed in for this option) to prevent a row break from being inserted within the group. Use in situations where you have 2 or more columns of related data you want to keep next to each other. For example, suppose you have three columns of data that repeat - date, time, and note. Set CELLGROUP=3 in this case so a row break will only occur after the 3rd (or 6th, or 9th, etc.) cell.
VARIABLECOLS	Optionally use this option with AUTOCOLS. Do not include TH (table header) rows with this option. This option tells Report Writer that the cells for each row may be a different size from the previous row. If you have various sized thumbnail images you wish to display, you might use this option to show more cells in a row containing smaller thumbnails.
PAGEROWS=number	Sets the maximum number of rows to print on a page.
YPOS=number	Conditionally page break based upon the current Y position when starting a new row. For example, if the current GRID settings are 100 in the Y direction and you set YPOS=70, the PAGE command will be ignored if the current Y position is less than 70. You may also enter the value followed by a % sign (i.e. YPOS=65%) to denote a percentage of the current GRID.
CELLSPLIT	Allows cells to split across a page boundary. Typically, all cell data in a given row for the outer most table must fit on the current page. A page break occurs and the entire row is moved to the next page if any one cell goes beyond the bottom page margin. This option prints what will fit of each cell on the current page and wraps the remaining text from each cell onto the next page or pages. You may only use this option on rows in the outer most table in the body of the report. This option is for cells containing text only (no embedded tables). Additionally, row options such as MINROWS and ROWSPAN should not be used with CELLSPLIT.

Tables

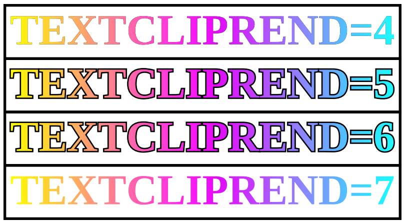
Parameter	Description
TBLALIGN=L|R|C	Sets the placement of the table between the X and X2 value to the Left (default), Center or Right.
NAME=text	Use the NAME parameter to keep the column widths of two or more tables sized the same. If you have a report where you are opening and closing a table several times due to page breaking or other circumstances and you want to keep a consistant look among these tables then use this parameter. You should only set the name of two or more tables to the same name when they have the same layout and are at the same nested level.
EXCEL=text	Set the worksheet (or tab) name you want this table exported to. Also used to start a new table when exporting to HTML. See the EXCEL tag for more information.
EXCEL_SEQ=number	Optional. Used to set the ordering sequence for the tab. Tabs are arranged in order from lower number to higher based on the sequence number, when supplied.
COMPTEXT	Compress text (when necessary) to fit in a cell rather than perform word wrapping.
DOTTEXT	Cuts off the end of any text that doesn't fit the cell and adds three dots (...) to the end of the text portion that does fit. Here is an example of COMPTEXT and DOTTEXT  Here is some text to show how it prints standard, compressed and dot-text. Here is some text to show how it prints standard, compressed and dot-text. Here is some text to show how i...
COMP=number	A percentage to compress the text by. A value less than 100 compresses text while a value greater than 100 expands text. Use this to override the compression setting the software may apply to wide tables.

Parameter	Description
BREAK=DOWN|OVER	Sets the way the table will break vertically if it's too wide for the page. The default method is to compress the text in the table so it will fit vertically on the page. Use this command to split the table instead and to specify the way in which the pages print. Setting break to DOWN will print down as far as necessary then go over to the remainder of the table. Here is the print sequence for BREAK=DOWN:  Setting break to OVER will print over as far as necessary then go down to the remainder of the table. Here is the print sequence for BREAK=OVER:  You can use the HEADER option on the TD cells to specify which columns are repeated on page breaks. There is also a PAGE option on TD cells to specify where the vertical breaks take place.

Parameter	Description
SOFTBREAK=number	Specify the amount of room in inches to leave to between table sections when vertically breaking the table. Vertical sections of a table print on separate pages without this option. This option should only be used for tables whose total height is less than half the page and the BREAK option described above is used. If you use this option on a table that breaks vertically with two small sections, for example, then both sections could be printed on the same page. Note that using this option with BREAK=OVER will automatically set the WHOLETABLE option.
WHOLETABLE	Print the table on the current page only if there is enough room. Otherwise, start the table on the top of the next page.
BACKGROUND=text	Set to an image file on disk. Use quotes around the file name. This option places the specified image on the page with the upper left corner of the image set at the upper left corner of the table. Table cells fall on top of the image so you can place text over the image. There is no other relation between the table and the image so it's possible to have a table that is larger or smaller than the image itself.
CELLBACKGROUND=text	Set to an image file on disk. Use quotes around the file name. This option sets the default for the BACKGROUND option in the row and cell tags.
CELLBACKGROUND2=text	Set to an image file on disk. Use quotes around the file name. This option sets the default for the BACKGROUND2 option in the row and cell tags.
TEXTCLIP=text	Set to an image file on disk. Use quotes around the file name. This option sets the default for the TEXTCLIP option in the row and cell tags.
TILE	Used to tile the background image specified by CELLBACKGROUND.
TILE2	Used to tile the background image specified by CELLBACKGROUND2.
TEXTCLIPTILE	Used to tile the background image specified by TEXTCLIP.

Tables

Parameter	Description
TEXTCLIPREND=number	Rendering mode to use with TEXTCLIP. Range is an integer from 4 to 7. 4 = Fill text then add to path for clipping 5 = Stroke text and add to path for clipping 6 = Fill, then stroke text and add to path for clipping 7 = Add text to path for clipping Default is 7. Modes 4-7 and 5-6 look the same since the image to clip will always fill the cell background.

	 TEXTCLIPREND=4
	TEXTCLIPREND=5
	TEXTCLIPREND=6
	TEXTCLIPREND=7

BORDER=number	Sets the border size. This option is used for a border around individual cells. A value of 1 is a line with a width of 1/72 of an inch. Default is 0 (no border).
BORDERCOLOR=color	Sets the border color (when the 3D option is not used).
BORDERDASH=number [,number,...]	A number or comma separated list of numbers to specify the ON/OFF length of the border.

Parameter	Description
BORDEROPTS=text	Used to supply border options in a shorthand format. Specify the border (top, bottom, left or right) followed by a colon then the size, type, and color. For example: "border:1 solid black" is used for a black border on all sides that is a width of 1 point. Use a semi-colon to separate multiple entries such as "border:1 solid black; border-bottom:2 dashed red;". This example will draw a solid 1 point black border on the top, left and right and a red dashed border that is 2 points thick on the bottom. You may use "border:" to set the options for all sides then use "border-top:", "border-bottom:", "border-left:", and "border-right:" to set any of the other borders as needed. Use a width of 0 to turn off a particular side, such as "border:1 solid black;border-top:0;border-bottom:1 dashed green". The type can be either solid, dotted, dashed, or a pair or numbers to represent the on/off length of a dash. The color can be any valid color setting.
CORNERSIZE=number	The size of the optional CORNERSTYLE in inches (centimeters if METRIC is used). Consider using CELLPADDING when setting this to a relatively large value so the text remains within the confines of the border.
CORNERSTYLE=text	The optional type of corners to use for the border. The default border is a simple rectangle. Valid values are: Round Bevel Scoop Stair

Parameter	Description
	In addition, you may specify a different style for any given corner. Pass a comma separated string containing a code for each corner followed by a colon then the type. The codes for each corner are: TL - Top left TR - Top right BR - Bottom right BL - Bottom left For example, set to "TL:round,TR:scoop,BR:round" to give the top left and bottom right corners a round edge and the top right a scoop edge. The bottom left edge, since it wasn't specified, will have a standard right-angled corner.
SIDES=text	Sets what sides to draw the border on. Default is all sides. Values are L (Left), R (Right), T (Top) and B (Bottom). For example, use SIDES=T,B to specify the border should be drawn on top and bottom only.
TABLEBORDER=number	Sets the border size. This option is used for a border around the entire table, not each cell. A value of 1 is a line with a width of 1/72 of an inch. Default is 0 (no border).
TABLEBORDERTOP	Draws the table border on top of the table. Normally the table border is in the background behind the table. This option is useful in certain situations, such as when placing a rounded table border over rows that are filled with a different color.
TABLEBORDERPADDING=number1, number2,number3,number4	Sets extra padding for the table border. The number1 is the top and rotate clockwise so number2 is the padding on the right, number3 on the bottom and number4 on the left. The value is in points (1/72 of an inch).
TABLEBORDERCOLOR=color	Sets the table border color. This setting applies to the border around the entire table, not each cell.
TABLEBORDERDASH= number[,number,...]	A number or comma separated list of numbers to specify the ON/OFF length of the border. This setting applies to the border around the entire table, not each cell.
TABLEBGCOLOR=color	Sets the table border background color. This setting applies to the border around the entire table, not each cell.

Tables

Parameter	Description
TABLECORNERSIZE=number	The size of the optional TABLECORNERSTYLE in inches (centimeters if METRIC is used). This setting applies to the border around the entire table, not each cell. The default is .03.
TABLECORNERSTYLE=text	The optional type of corners to use for the border around the entire table. Valid values are: Round Bevel Scoop Stair In addition, you may specify a different style for any given corner. Pass a comma separated string containing a code for each corner followed by a colon then the type. The codes for each corner are: TL - Top left TR - Top right BR - Bottom right BL - Bottom left For example, set to "TL:round,TR:scoop,BR:round" to give the top left and bottom right corners a round edge and the top right a scoop edge. The bottom left edge, since it wasn't specified, will have a standard right-angled corner. This setting applies to the border around the entire table, not each cell.
TABLEPATTERN=number	A pattern number to use for the background. See the RECT command and its PATTERN option for a sample of each pattern's appearance. This setting applies to the border around the entire table, not each cell.
TABLEPATTERNCOLOR=color	Color for the pattern. This setting applies to the border around the entire table, not each cell.
TABLESIDES=text	Sets what sides to draw the table border on. Default is all sides. Values are L (Left), R (Right), T (Top) and B (Bottom). For example, use SIDES=T,B to specify the border should be drawn on top and bottom only.

Tables

Parameter	Description
TABLEFILL=color	Color for the table background. This setting applies to the background of the entire table. Use this option to prevent a background PDF from showing through on the table.
BORDERCOLS=list	Use this option with the TABLEBORDER option. Set to the list of from and thru column numbers (starting at 1 on the left) to draw the table border around. Using this option will draw boxes around a column or group of columns rather than the whole table. For example, BORDERCOLS=2,3,5,8 will draw two rectanges - one that covers columns 2 and 3 and another that covers columns 5 through 8.
TABLEBACKGROUND=text	Set to an image file on disk. Use quotes around the file name. The image is streched vertically and/or horizontally as needed to fill the background area of the table unless TILEBACKGROUND is used.
TILEBACKGROUND	Used to tile the background image specified by TABLEBACKGROUND.
BGCOLOR=color	Sets the background color of the table cells.
STRIPE=color,color	Set to a pair of colors to stripe (also called a zebra pattern) the rows in the table. The first row is the first color, the second row is colored the second color, the third the first color and so on. For example, STRIPE="#9cf,#fff" or STRIPE="red,blue". Use BGCOLOR on any TR row or TD cell where you want to override the setting. Also use STRIPEOFF or STRIPEON on any TR rows where you want to turn striping on or off.
STRIPEX=number,number	Forces the striping for each row to start and end at the percentages specified of the total table width. For example, specify 0,100 to span the entire table width. Use this in situations where you have rows that do not include all of the cells in the table.
SHADING=text	A shading pattern to use for the table cells. Set to the NAME value from the SHADING tag.

Parameter	Description
PATTERN=number	A pattern number to use for the background. See the RECT command and its PATTERN option for a sample of each pattern's appearance.
PATTERNCOLOR=color	Color for the pattern.
3D	Sets the border style as 3D which is lighter on top and left and darker on bottom and right. The base color used is the BGCOLOR or BORDERCOLOR if BGCOLOR is not specified.
COLORSTARTx=color	Replace the x in COLORSTARTx with the number of a column in the table. The leftmost column is 1, next one to the right is 2 and so on. This color is the starting font color used for the lowest value in the table column. Use COLORENDx along with this option. The text color will change based on the cell value as a percentage of the highest and lowest values in the column. Values closer to 0% will be colored similar to COLORSTART while numbers closer to 100% will be colored similar to COLOREND. Values around 50% will be colored similar to COLORMID, if COLORMID is used. See the example following this section.
COLORMIDx=color	Replace the x in COLORMIDx with the number of a column in the table. This is an optional middle color used in between COLORSTARTx and COLORENDx.
COLORENDx=color	Replace the x in COLORENDx with the number of a column in the table. This color is the ending font color used for the highest value in the table column. Used with COLORSTARTx.

Tables

Parameter	Description
COLORBGSTARTx=color	Replace the x in COLORBGSTARTx with the number of a column in the table. The leftmost column is 1, next one to the right is 2 and so on. This color is the starting cell background color used for the lowest value in the table column. Use COLORBGENDx along with this option. The background cell color will change based on the cell value as a percentage of the highest and lowest values in the column. Values closer to 0% will be colored similar to COLORBGSTART while numbers closer to 100% will be colored similar to COLORBGEND. Values around 50% will be colored similar to COLORBGMID, if COLORBGMID is used. See the example following this section. Also see the COLORBGCONTENTSx option.
COLORBGMIDx=color	Replace the x in COLORBGMIDx with the number of a column in the table. This is an optional middle color used in between COLORBGSTARTx and COLORBGENDx.
COLORBGENDx=color	Replace the x in COLORBGENDx with the number of a column in the table. This color is the ending cell background color used for the highest value in the table column. Used with COLORBGSTARTx.
COLORBGCONTENTSx	Replace the x in COLORBGCONTENTSx with the number of a column in the table. Treats COLORBGSTARTx as a BGCOLOR tag within the cell rather than a BGCOLOR assigned on the TD cell itself. That is, the printable contents are colored in rather than the default of the entire cell. For example, you might use this when using MULTISORT on the table.
COLORBARx=number	Replace the x in COLORBARx with the number of a column in the table. Displays a horizontal bar in the cell rather than the number. The number passed is the height of the bar in points (1/72 of an inch). The color of the bar is based on the COLORBGSTARTx and COLORBGENDx settings. You may have the same color set for both COLORBGSTARTx and COLORBGENDx if the bar is to be the same color regardless of the length.

Tables

Parameter	Description
COLORBARROUNDx=number	Replace the x in COLORBARROUNDx with the number of a column in the table. Creates a rounded edge on the horizontal bar. Pass in the width of the rounded edge in points (1/72 of an inch). Used with COLORBARx.
COLORBARBLOCKx=number	Replace the x in COLORBARBLOCKx with the number of a column in the table. Creates a series of blocks rather than a solid bar. Pass in the block spacing of the in points (1/72 of an inch). Used with COLORBARx.
COLORBARIMGx=text	Replace the x in COLORBARIMGx with the number of a column in the table. Pass in the path-name of an image file. The image specified is used for drawing the bar. The image is scaled to fit (when necessary) and tiled. Used with COLORBARx.
COLORBARTOPx=number	Replace the x in COLORBARTOPx with the number of a column in the table. An offset in points (1/72 of an inch) from the top of the cell for the top of the bar.
COLORBARBCx=color	Replace the x in COLORBARBCx with the number of a column in the table. A color for the border of all bars drawn for the column. For use with COLORBARx.
COLORBARZEROx=number	Replace the x in COLORBARZEROx with the number of a column in the table. A number representing a percent from 0 to 100 to base the zero position of the bar at. For example, suppose the range of values for a column goes from 0 to 20. If you set COLBARZEROx to 50, values under 10 will be drawn from the center of the bar to the left. Values close to zero will be drawn almost to the far left of the bar and extent to the middle. Values greater than 10 will be drawn from the center to the right. A value of 10 in this example will simply be a line in the middle of the bar. For use with COLORBARx.

Tables

Parameter	Description
COLORBGMINx=number	Replace the x in COLORBGMINx with the number of a column in the table. For use with COLORSTARTx or COLORBGSTARTx. Used to specify the minimum value for the column. This is optional for when you want to set the minimum value to something other than the lowest value found in the table column. For example, if you want your color range to go from 0 to 100 then set this to 0. If the lowest value in the column is 40 it will then be treated as 40% (on the 0 to 100 scale) rather than the lowest percent value of 0%.
COLORBGMAXx=number	Replace the x in COLORBGMAXx with the number of a column in the table. For use with COLORSTARTx or COLORBGSTARTx. Used to specify the maximum value for the column. This is optional for when you want to set the maximum value to something other than the highest value found in the table column. For example, if you want your color range to go from 0 to 100 then set this to 100. If the highest value in the column is 80 it will then be treated as 80% (on the 0 to 100 scale) rather than the highest percent value of 100%.
COLSORT=text	A column number to sort the table on. Place an N after the column number to specify a numeric sort - for when your column contains a set of numbers. For example, "3N" means sort the table on column 3 numerically. The default sort is alpha-numeric.
COLSORTDIR=text	The direction of the column sort. Use D for descending or A (the default) for ascending.
BREAKONCHANGE=number	Enter a column number (starting from 1 for the leftmost column) to check for a change in the contents. Once the contents have changed in the column, a page break will be inserted. For example, use this to page break an employee listing by department. If department is in the second column, set BREAKONCHANGE=2. A page break will occur each time a new department is found.

Tables

Parameter	Description
MULTISORT=list	Assign a list of columns to provide ascending and descending sorts on. This feature uses layers so Acrobat or Reader 6.0 or higher is required. Earlier versions will simply display all of the layers on top of one another and thus be unreadable. For example, to provide a table that the end user can sort from lowest to highest on columns 1 or 3, pass "1,3" as the value. The sort is alpha-numeric by default. Place an N after the column number to specify a numeric sort - for when your column contains a set of numbers. For example, "1,3N" means sort column 1 alpha-numerically and column 3 numerically. You must specify the MULTISORTDESC option as well for this option to work. The first column listed is the default sort regardless of how you passed the data to Report Writer. So, if you want the table to start off with column 3 sorted by default, use "3N,1" as the list. Use the MULTISORTARROWS option and click on the up or down arrow in the column heading. Also, or instead, open the "Layers" tab in Acrobat or Reader to see the layers. The appropriate layer is brought into view when you click on one, thus allowing dynamic sorts on the table by the end user. Only use this option on simple tables not containing column/row spans or embedded tables. Individual TD cell settings options, such as BGCOLOR, must be the same on all cells in a given column.

Parameter	Description
MULTISORTDESC=list	Pass in a list of descriptions to use for the sort. Each column will be sorted ascending and descending so you'll need to pass two entries for each MULTISORT entry. For example, assume MULTISORT is set to "1,3N". If column 1 is a list of names and column 3 contains salary, you'd want to set MULTISORTDESC to something like "Name (Asc),Name (Desc),Salary (Asc),Salary (Desc)". The ascending description must come first as that is how it will be ordered in the layer name list. Keep the list of descriptions unique throughout a document when using more than one table. Otherwise, all tables will matching descriptions be sorted as a group.
MULTISORTARROWS	Creates an up and down arrow in the column headings (the TH table heading tag) being sorted. May want to add a <BR> after the text if you want the arrows on the next line. Otherwise, they'll appear right after whatever heading text you have. You may then click on the up or down arrow in the appropriate column to sort the table.
MULTISORTSIZE=number	Optional. The point size for the arrows when using MULTISORTARROWS. The default is the point size used in the cell.
MULTISORTUPCOLOR=text	The color of the up arrow when using MULTISORTARROWS. The default is blue.
MULTISORTDOWNCOLOR=text	The color of the down arrow when using MULTISORTARROWS. The default is blue.
MULTISORTUPIMG=text	Optional. The path and name of an image to use in place of the up ▲ arrow. Must include the MULTISORTARROWS option as well.
MULTISORTDOWNIMG=text	Optional. The path and name of an image to use in place of the down ▼ arrow. Must include the MULTISORTARROWS option as well.

Tables

Parameter	Description
COLNUM=number	Sets the number of columns on the page to divide the entire table into. Setting to two, for example, will fill up the left half of the page then move to the top right half of the page and continue. A new page is started automatically once all page columns are filled. An example of this would be a phone book which typically has two table columns, name and number, which wrap two or three times on a page before starting a new page. Keep in mind this is the number of columns to divide the page into, not the number of columns in the underlying table.
COLSPACE=number	Sets the amount of space in units based on the GRID command between each page column.
COLBAR=number	Draws a vertical bar between multiple page columns. Set this parameter to the line width.
ALIGN=L|R|C|J	Sets the default alignment for the table to Left (default unless overridden by BASEALIGN command), Center, Right or Justified.
VALIGN=T|M|B	Sets the default vertical alignment for the table to Top (default), Middle or Bottom. Also see the VALIGN2 option. The BASELINE option can be used to set the point to use for middle alignment if you want something different than the middle of the text block.
FACE=font number	Sets the font.
SIZE=point size	Sets the point size for the font.
LINESPACE=number	Sets the text line spacing. Each unit is 1/72 of an inch. Default is 2.
FCOLOR=color	Sets the fill color. This is also the font color.
SCOLOR=color	Sets the stroke color.
CONTENTANGLE=number	Pass in the number of degrees (0-360) to rotate cell contents counter-clockwise. Or use a negative number from -360 to 0 to rotate cell contents clockwise. Only the contents of the cell are rotated - not the border or cell background. This sets the default for all cells in the table.

Tables

Parameter	Description
CONTENTANGLES=number	Optional scaling factor to use with CONTENTANGLE. For example, pass in 50 for 50%.
CONTENTBORDER=number	Sets the width in units of 1/72 of an inch of a border drawn around the cell contents. The cell contents is the area of text or graphics drawn inside of a cell. This area can be significantly smaller than the cell size. This border is rotated with the CONTENTANGLE command.
CONTENTBORDERCOLOR=color	The color of the content border.
CONTENTBORDERBGCOLOR=color	The background color of the content border.
CONTENTBORDERPADX=number	Sets extra horizontal padding for the content border based on units of 1/72 of an inch. You may use decimal values.
CONTENTBORDERPADY=number	Sets extra vertical padding for the content border based on units of 1/72 of an inch. You may use decimal values.
FLUSH	Ignore column padding for inner tables. Inner tables are flush against the outer table.
QUICK	Builds vertically breaking tables faster. Has no affect on tables that are not broken vertically. Only use this option when each row will contain the same number of text lines. This way the software doesn't need to check the height of cells in the same row not printed on the same page.
FORCEWIDTH	Forces the program to use the column widths specified on TD tags rather than expand to fit the maximum of the cell contents vs. the width setting.
BREAKON="text"	Set of characters a line break may occur on in addition to the space character. This overrides any global setting made with the BREAKON tag.
OL_LEVEL=number	The outline level.
OL_DESCR=text	The text of the outline (leave blank if you are using the table column without a description).
OL_CLOSED	Set this option to initially close the outline.
OL_SUB_LEVEL=number	The sub outline level.

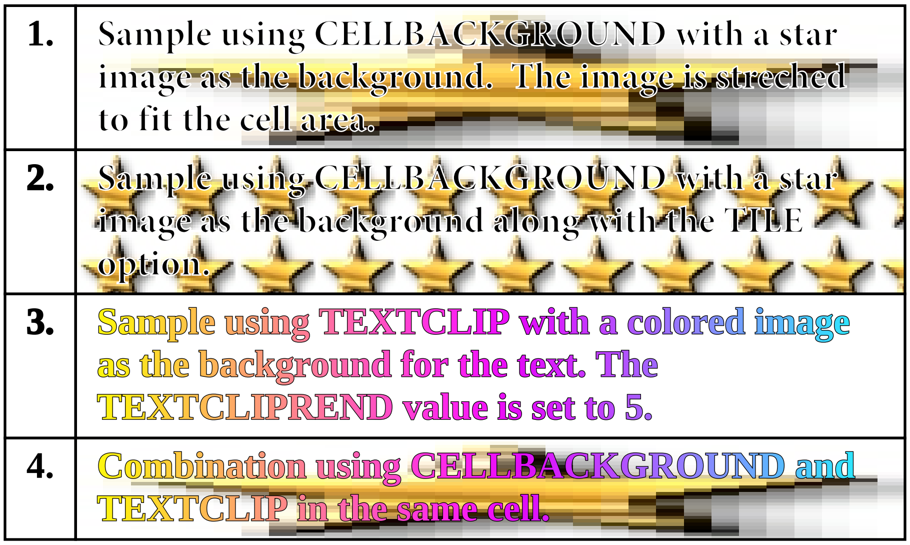
Parameter	Description
OL_SUB_CLOSED	Set this option to initially close the sub outline.
OL_COL=number	The column number to use for the index (left most column is 1). You may also use a string that references the colums as &1 for column 1, &2 for column 2, etc. For example, "&1 (&2)" or "&1 - &2" where column 1 is a main account or part number and column 2 is a sub number or description.
OL_ALL	Use all column entries in the outline, not just the first one on the page. Each row in the table becomes an outline entry in this case.
OL_BREAK	A character to break on for the outline text. Only text up to the first occurrence of the break character is used. For example, if the text "Standard Size (3 x 15)" is the cell text for the outline and OL_BREAK is set to "(" then only the text "Standard Size" will be used as the outline text.
OL_LENGTH	The length to use for the outline text. For example, if the text "125-A500 - Copper" is the cell text for the outline and OL_LENGTH is set to 8 then only the text "125-A500" will be used as the outline text.
PLAIN[=text]	An optional text character to use as the sizing character. The character "1" is used as the default. See the PLAIN global command for more information. Use PLAIN="" if you set the global PLAIN command and wish to turn it off for the given TABLE. Note that all formatting is ignored when using this option.

The OL_ parameters are used to set up bookmark entries based on a column in the table. You may specify a column as the index which, in order for it to make sense, should be column that the table is sorted on.

To create a table that will display information on customers sorted by customer name in the second column you might set up the outline as follows:
OL_LEVEL=1
OL_DESCR="Customer Name"
OL_CLOSED
OL_SUB_LEVEL=2
OL_COL=2

To just create the index without a heading, set up the outline like this:
OL_LEVEL=1
OL_COL=2

CELLBACKGROUND/TEXTCLIP example:



1.	Sample using CELLBACKGROUND with a star image as the background. The image is streched to fit the cell area.
2.	Sample using CELLBACKGROUND with a star image as the background along with the TILE option.
3.	**Sample using TEXTCLIP with a colored image as the background for the text. The TEXTCLIPREND value is set to 5.**
4.	**Combination using CELLBACKGROUND and TEXTCLIP in the same cell.**

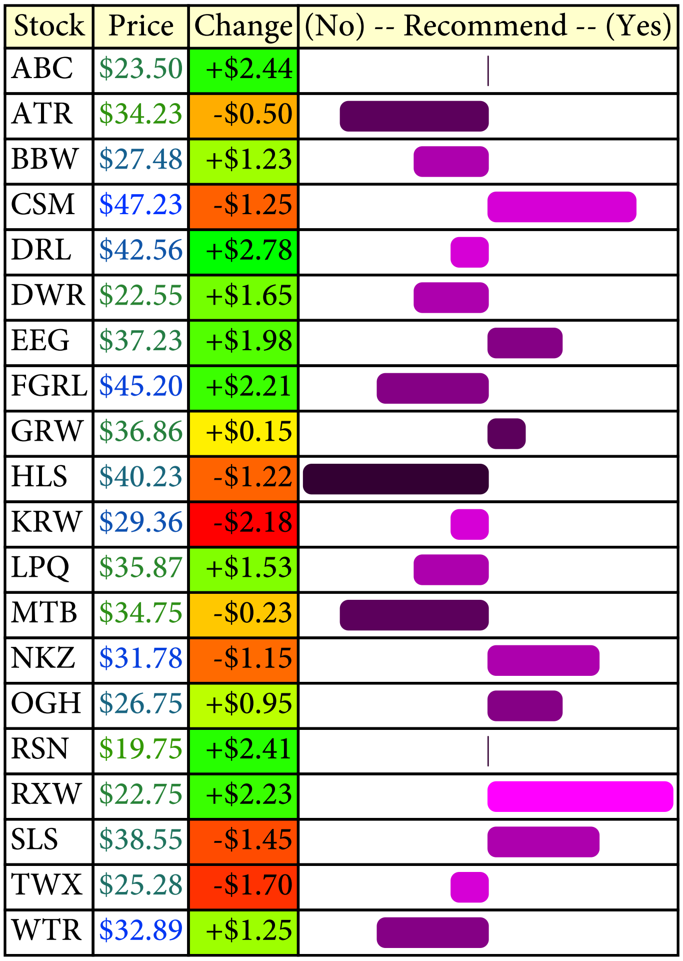
COLORSTART example (be sure to use COLORBGCONTENTS if using this with MULTISORT). The example source is several pages and the output is shown on the page following:

```
<ALIGN VALUE=L>
<TABLE BORDER=1 CELLPADDING=2 SIZE=12
 COLORSTART2=#339900 COLOREND2=#0033FF
 COLORBGSTART3=#FF0000 COLORBGMID3=#FFFF00 COLORBGEND3=#00FF00
 COLORBGSTART4=#3300FF COLORBGEND4=#FF00FF
 COLORBAR4=11 COLORBARTOP4=2
 COLORBARZERO4=50 COLORBARROUND4=3>
<TH BGCOLOR=#FFFFCC ALIGN=C COLORSPAN_OFF>
<TD>Stock</TD>
<TD>Price</TD>
<TD>Change</TD>
<TD>(No) -- Recommend -- (Yes)</TD>
</TH>
<TR>
<TD>ABC</TD>
<TD ALIGN=R>$23.50</TD>
<TD ALIGN=R>+$2.44</TD>
<TD>5</TD>
</TR>
<TR>
<TD>ATR</TD>
<TD ALIGN=R>$34.23</TD>
<TD ALIGN=R>-$0.50</TD>
<TD>1</TD>
</TR>
<TR>
<TD>BBW</TD>
<TD ALIGN=R>$27.48</TD>
<TD ALIGN=R>+$1.23</TD>
<TD>3</TD>
</TR>
<TR>
<TD>CSM</TD>
<TD ALIGN=R>$47.23</TD>
<TD ALIGN=R>-$1.25</TD>
<TD>9</TD>
</TR>
<TR>
<TD>DRL</TD>
<TD ALIGN=R>$42.56</TD>
<TD ALIGN=R>+$2.78</TD>
<TD>4</TD>
</TR>
<TR>
<TD>DWR</TD>
<TD ALIGN=R>$22.55</TD>
<TD ALIGN=R>+$1.65</TD>
```

```
<TD>3</TD>
</TR>
<TR>
<TD>EEG</TD>
<TD ALIGN=R>$37.23</TD>
<TD ALIGN=R>+$1.98</TD>
<TD>7</TD>
</TR>
<TR>
<TD>FGRL</TD>
<TD ALIGN=R>$45.20</TD>
<TD ALIGN=R>+$2.21</TD>
<TD>2</TD>
</TR>
<TR>
<TD>GRW</TD>
<TD ALIGN=R>$36.86</TD>
<TD ALIGN=R>+$0.15</TD>
<TD>6</TD>
</TR>
<TR>
<TD>HLS</TD>
<TD ALIGN=R>$40.23</TD>
<TD ALIGN=R>-$1.22</TD>
<TD>0</TD>
</TR>
<TR>
<TD>KRW</TD>
<TD ALIGN=R>$29.36</TD>
<TD ALIGN=R>-$2.18</TD>
<TD>4</TD>
</TR>
<TR>
<TD>LPQ</TD>
<TD ALIGN=R>$35.87</TD>
<TD ALIGN=R>+$1.53</TD>
<TD>3</TD>
</TR>
<TR>
<TD>MTB</TD>
<TD ALIGN=R>$34.75</TD>
<TD ALIGN=R>-$0.23</TD>
<TD>1</TD>
</TR>
<TR>
<TD>NKZ</TD>
<TD ALIGN=R>$31.78</TD>
<TD ALIGN=R>-$1.15</TD>
<TD>8</TD>
</TR>
<TR>
```

```
<TD>OGH</TD>
<TD ALIGN=R>$26.75</TD>
<TD ALIGN=R>+$0.95</TD>
<TD>7</TD>
</TR>
<TR>
<TD>RSN</TD>
<TD ALIGN=R>$19.75</TD>
<TD ALIGN=R>+$2.41</TD>
<TD>5</TD>
</TR>
<TR>
<TD>RXW</TD>
<TD ALIGN=R>$22.75</TD>
<TD ALIGN=R>+$2.23</TD>
<TD>10</TD>
</TR>
<TR>
<TD>SLS</TD>
<TD ALIGN=R>$38.55</TD>
<TD ALIGN=R>-$1.45</TD>
<TD>8</TD>
</TR>
<TR>
<TD>TWX</TD>
<TD ALIGN=R>$25.28</TD>
<TD ALIGN=R>-$1.70</TD>
<TD>4</TD>
</TR>
<TR>
<TD>WTR</TD>
<TD ALIGN=R>$32.89</TD>
<TD ALIGN=R>+$1.25</TD>
<TD>2</TD>
</TR>
</TABLE>
```

Tables



Stock	Price	Change	(No) -- Recommend -- (Yes)
ABC	$23.50	+$2.44	
ATR	$34.23	-$0.50	
BBW	$27.48	+$1.23	
CSM	$47.23	-$1.25	
DRL	$42.56	+$2.78	
DWR	$22.55	+$1.65	
EEG	$37.23	+$1.98	
FGRL	$45.20	+$2.21	
GRW	$36.86	+$0.15	
HLS	$40.23	-$1.22	
KRW	$29.36	-$2.18	
LPQ	$35.87	+$1.53	
MTB	$34.75	-$0.23	
NKZ	$31.78	-$1.15	
OGH	$26.75	+$0.95	
RSN	$19.75	+$2.41	
RXW	$22.75	+$2.23	
SLS	$38.55	-$1.45	
TWX	$25.28	-$1.70	
WTR	$32.89	+$1.25	

SHADING example. Use the SHADING tag to create a 3D like appearance within your table cells.

```
<SHADING NAME="metal"
  COLOR1=.75,.75,.75 COLOR2=white COLORARY="0,1,0,0">

<SHADING NAME="navy"
  COLOR1=navy COLOR2=#9999FF COLORARY="1,1,0,0">

<TABLE BORDER=1 CELLPADDING=2 SIZE=12 SHADING="metal">

<TH SHADING="navy" FCOLOR=white>
<TD>Company</TD>
<TD>Country</TD>
</TH>

<TR>
<TD>British Consolidated</TD>
<TD>United Kingdom</TD>
</TR>

<TR>
<TD>General Motors</TD>
<TD>United States</TD>
</TR>

<TR>
<TD>GRB International</TD>
<TD>Germany</TD>
</TR>

<TR>
<TD>Nintendo</TD>
<TD>Japan</TD>
</TR>

</TABLE>
```

Company	Country
British Consolidated	United Kingdom
General Motors	United States
GRB International	Germany
Nintendo	Japan

MULTISORT example (must use Acrobat or Reader 6.0 or higher to view properly):

```
<ALIGN VALUE=L>
<TABLE BORDER=1 CELLPADDING=2 SIZE=12 MULTISORT=1,3N
  MULTISORTDESC="Name asc,Name desc,Salary asc,Salary desc"
  MULTISORTARROWS>
<TH BGCOLOR=#FFFFCC ALIGN=C>
<TD>Name </TD><TD>Title</TD><TD>Salary </TD>
</TH>
<TR>
<TD MULTISORTVAL="Harris,Robert">Robert Harris</TD>
<TD>Manager</TD>
<TD ALIGN=R>$56,000</TD>
</TR><TR>
<TD MULTISORTVAL="Smith,Joe">Joe Smith</TD>
<TD>Manager</TD>
<TD ALIGN=R>$52,500</TD>
</TR><TR>
<TD MULTISORTVAL="Zimmer,Jane">Jane Zimmer</TD>
<TD>CEO</TD>
<TD ALIGN=R>$135,000</TD>
</TR><TR>
<TD MULTISORTVAL="Forest,Jack">Jack Forest</TD>
<TD>Sales Rep</TD>
<TD ALIGN=R>$38,000</TD>
</TR><TR>
<TD MULTISORTVAL="Beech,Kelly">Kelly Beech</TD>
<TD>Sales Rep</TD>
<TD ALIGN=R>$40,000</TD>
</TR>
<TR NO_MULTISORT>
<TD>Total</TD>
<TD></TD>
<TD ALIGN=R>$265,500</TD>
</TR>
</TABLE>
```

Click on an arrow in the header to sort the data. Name column is using last name for sort.

Name ▲ ▼	Title	Salary ▲ ▼
Kelly Beech	Sales Rep	$40,000
Jack Forest	Sales Rep	$38,000
Robert Harris	Manager	$56,000
Joe Smith	Manager	$52,500
Jane Zimmer	CEO	$135,000
Total		$265,500

<LEGEND
    [...options...]>
</LEGEND>

Used to place a legend (typically a short text string) at the top of the table. This text is outside of any table row or cell and no extra space is allocated for it, meaning you might want to put a BR tag before the start of the table if want to allow extra room. The legend is simply a small description or title for the table. The text of the legend goes between the opening <LEGEND> tag and the closing </LEGEND>.

This tag has the same options as the TEXT tag. The only difference is the X and Y options on the LEGEND tag are an offset (in points or 1/72 of an inch) from the top of the table. Positive X values move the text to the left, negative to the right. Positive Y values move the text down, negative moves it up.

You may want to use the BGCOLOR and/or BORDER options to provide a background for the text. This tag should go after the TABLE tag and before the first TBODY or TH/TF/TR tag. For instance:

```
<TABLE>
 <LEGEND BORDER=1 BGCOLOR=#CCCCCC X=15 Y=2 SIZE=15>
 Company Assets
 </LEGEND>
 <TH>
  <TD>...Header cells...</TD>
 </TH>
 <TR>
  <TD>...</TD>
 </TR>
 <TR>
  <TD>...</TD>
 </TR>
 <TR>
  <TD>...</TD>
 </TR>
</TABLE>
```

<TBODY
[*...options...*]>
</TBODY>

Used to apply a layout or setting to group of rows. You may include more than one TBODY tag per table but they should not be nested.

For example, to change the background color of a set of ROWS, you can place a BGCOLOR option on each TR tag. Or, add a TBODY tag around the TR rows and set BGCOLOR on TBODY instead. Any options set in the TR row itself will override the TBDOY setting. For instance:

```
<TABLE>
 <TH>
  <TD>...Header cells...</TD>
 </TH>
 <TBODY BGCOLOR=red BORDER=1>
   <TR>
    <TD>...</TD>
   </TR>
   <TR>
    <TD>...</TD>
   </TR>
 </TBODY>
 <TR>
  <TD>...</TD>
 </TR>
</TABLE>
```

The options for TBODY are the same as the TR tag.

```
<TH or <TF
    BORDER=number
    BORDERCOLOR=color
    BORDERDASH=number
      [,number,...]
    BORDEROPTS=text
    CORNERSIZE=number
    CORNERSTYLE=text
    BGCOLOR=color
    SHADING=text
    PATTERN=number
    PATTERNCOLOR=color
    3D
    ROWBORDER
    SIDES=text
    LINE_OVER=number
    LINE_OVER_OFFSET=number
    LINE_UNDER=number
    LINE_UNDER_OFFSET=number
    CELLBACKGROUND=text
    CELLBACKGROUND2=text
    TEXTCLIP=text
    TILE
    TILE2
    TEXTCLIPTILE
    TEXTCLIPREND=number
    ALIGN=L|R|C|J
    VALIGN=T|M|B
    HEIGHT=number
    FACE=font number
    SIZE=point size
    COMP=number
    COMPTEXT[=OFF]
    DOTTEXT
    LINESPACE=number
    CELLXPADDING=number
    FCOLOR=color
    SCOLOR=color
    CONTENTANGLE=number
    CONTENTANGLES=number
    CONTENTBORDERCOLOR=color
    CONTENTBORDERBGCOLOR=color
    CONTENTBORDERPADX=number
    CONTENTBORDERPADY=number
    COLORSPAN_OFF
    LAYER=name
    LAYER_OFF>
```

</TH> or </TF>

Used to define a table header or footer row. Use TH for a table header or TF for a table footer. If using either one (or both) of these, the TH or TF rows must be placed before a standard TR row. For instance:

```
<TABLE>
 <TH>
  <TD>...Header cells...</TD>
 </TH>
 <TF>
  <TD>...Footer cells...</TD>
 </TF>
 <TR>
  <TD>...</TD>
 </TR>
 <TR>
  <TD>...</TD>
 </TR>
</TABLE>
```

These tags appear inside of the <TABLE> </TABLE> tags. You may have as many header or footer rows as you want. For tables that span pages, these rows are repeated at the top of each new page. Like the table row (<TR> tag) a table header or footer will not split across a page. If a new table is starting at the bottom of a page and there is enough room for the header but no data rows, the header will not print at the bottom of the page.

You can also use <THEAD><TR> in place of <TH> and close it with </TR></THEAD>. In the same way, you can use <TFOOT><TR> in place of <TF> and close it with </TR></TFOOT>. For example:

```
<TABLE>
 <THEAD>
 <TR>
  <TH>Header cell</TH>
 </TR>
 </THEAD>
 <TFOOT>
 <TR>
  <TH>Footer cell</TH>
 </TR>
 </TFOOT>
 <TR>
  <TD>Table Cell 1</TD>
 </TR>
 <TR>
  <TD>Table Cell 2</TD>
 </TR>
</TABLE>
```

The TH functions as a TD when used within a THEAD or TFOOT section.

All of the parameters are optional. Any values not set default to the values set in the TABLE tag.

Parameter	Description
BORDER=number	Sets the border size. A value of 1 is a line with a width of 1/72 of an inch.
BORDERCOLOR=color	Sets the border color (when the 3D option is not used).
BORDERDASH=number [,number,...]	A number or comma separated list of numbers to specify the ON/OFF length of the border.
BORDEROPTS=text	Used to supply border options in a shorthand format. Specify the border (top, bottom, left or right) followed by a colon then the size, type, and color. For example: "border:1 solid black" is used for a black border on all sides that is a width of 1 point. Use a semi-colon to separate multiple entries such as "border:1 solid black; border-bottom:2 dashed red;". This example will draw a solid 1 point black border on the top, left and right and a red dashed border that is 2 points thick on the bottom. You may use "border:" to set the options for all sides then use "border-top:", "border-bottom:", "border-left:", and "border-right:" to set any of the other borders as needed. Use a width of 0 to turn off a particular side, such as "border:1 solid black;border-top:0;border-bottom:1 dashed green". The type can be either solid, dotted, dashed, or a pair or numbers to represent the on/off length of a dash. The color can be any valid color setting.
CORNERSIZE=number	The size of the optional CORNERSTYLE in inches (centimeters if METRIC is used). Consider using CELLPADDING when setting this to a relatively large value so the text remains within the confines of the border.
CORNERSTYLE=text	The optional type of corners to use for the border. The default border is a simple rectangle. Valid values are: Round Bevel Scoop Stair

Tables

Parameter	Description
	In addition, you may specify a different style for any given corner. Pass a comma separated string containing a code for each corner followed by a colon then the type. The codes for each corner are: TL - Top left TR - Top right BR - Bottom right BL - Bottom left For example, set to "TL:round,TR:scoop,BR:round" to give the top left and bottom right corners a round edge and the top right a scoop edge. The bottom left edge, since it wasn't specified, will have a standard right-angled corner.
BGCOLOR=color	Sets the background color of the row.
SHADING=text	A shading pattern to use for the row. Set to the NAME value from the SHADING tag.
PATTERN=number	A pattern number to use for the background. See the RECT command and its PATTERN option for a sample of each pattern's appearance.
PATTERNCOLOR=color	Color for the pattern.
3D	Sets the border style as 3D which is lighter on top and left and darker on bottom and right. The base color used is the BGCOLOR or BORDERCOLOR if BGCOLOR is not specified.
ROWBORDER	Draws the border around the entire row rather than each cell. Do not use on rows containing cells with the ROWSPAN option.
SIDES=text	Sets what sides to draw the border on. Default is all sides. Values are L (Left), R (Right), T (Top) and B (Bottom). For example, use SIDES=T,B to specify the border should be drawn on top and bottom only.
LINE_OVER=number	Draws a line above each cell in the row. Set number to 1 for a single line or 2 for a double line. In contrast to a border, this line does not extend into the cellpadding around each cell. This option can be used to draw a line for a row of totals for instance.
LINE_OVER_OFFSET=number	An offset along the Y-axis in points (1/72 of an inch) to use with LINE_OVER. Positive numbers move the line down while negative numbers move the line up.

Tables

Parameter	Description
LINE_UNDER=number	Draws a line below each cell in the row. Set number to 1 for a single line or 2 for a double line. In contrast to a border, this line does not extend into the cellpadding around each cell. This option can be used to underline a heading row for instance.
LINE_UNDER_OFFSET=number	An offset along the Y-axis in points (1/72 of an inch) to use with LINE_UNDER. Positive numbers move the line down while negative numbers move the line up.
CELLBACKGROUND=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for all the cells in the row. The image is streched vertically and/or horizontally as needed to fill the area unless TILE is used. This option fills the cell background but not the area included in any cellpadding.
CELLBACKGROUND2=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for all the cells in the row. The image is streched vertically and/or horizontally as needed to fill the area unless TILE2 is used. This option fills the cell background including the area used for cellpadding.
TEXTCLIP=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for the text characters in the row. The image is streched vertically and/or horizontally as needed to fill the cell area unless TEXTCLIPTILE is used.
TILE	Used to tile the background image specified by CELLBACKGROUND.
TILE2	Used to tile the background image specified by CELLBACKGROUND2.
TEXTCLIPTILE	Used to tile the background image specified by TEXTCLIP.
TEXTCLIPREND=number	Rendering mode to use with TEXTCLIP. Range is an integer from 4 to 7. Default is 7.
ALIGN=L|R|C|J	Sets the default alignment for the row to Left (default), Center, Right or Justified.

Tables

Parameter	Description
VALIGN=T|M|B	Sets the default vertical alignment for the row to Top (default), Middle or Bottom. Also see the VALIGN2 option. The BASELINE option can be used to set the point to use for middle alignment if you want something different than the middle of the text block.
HEIGHT=number	Forces the row to be at least as high in YUNITS as the number specified.
FACE=font number	Sets the font.
SIZE=point size	Sets the point size for the font.
COMP=number	A percentage to compress the text by. A value less than 100 compresses text while a value greater than 100 expands text.
COMPTEXT[=OFF]	Compress text (when necessary) to fit in a cell rather than perform word wrapping. Use COMPTEXT=OFF to turn option off for the current row.
DOTTEXT	Cuts off the end of any text that doesn't fit the cell and adds three dots (...) to the end of the text portion that does fit.
LINESPACE=number	Sets the text line spacing. Each unit is 1/72 of an inch. Default is 2.
CELLXPADDING=number	Sets extra horizontal padding within cells based on units of 1/72 of an inch. You may use decimal values. This is only used when drawing the border.
FCOLOR=color	Sets the fill color. This is also the font color.
SCOLOR=color	Sets the stroke color.
CONTENTANGLE=number	Pass in the number of degrees (0-360) to rotate cell contents counter-clockwise. Or use a negative number from -360 to 0 to rotate cell contents clockwise. Only the contents of the cell are rotated - not the border or cell background. This sets the default for all cells in the row.
CONTENTANGLES=number	Optional scaling factor to use with CONTENTANGLE. For example, pass in 50 for 50%.
CONTENTBORDER=number	Sets the width in units of 1/72 of an inch of a border drawn around the cell contents. The cell contents is the area of text or graphics drawn inside of a cell. This area can be significantly smaller than the cell size. This border is rotated with the CONTENTANGLE command.

Tables

Parameter	Description
CONTENTBORDERCOLOR=color	The color of the content border.
CONTENTBORDERBGCOLOR=color	The background color of the content border.
CONTENTBORDERPADX=number	Sets extra horizontal padding for the content border based on units of 1/72 of an inch. You may use decimal values.
CONTENTBORDERPADY=number	Sets extra vertical padding for the content border based on units of 1/72 of an inch. You may use decimal values.
COLORSPAN_OFF	Turns off any COLORSTARTx or COLORBGSTARTx tags for the row.
LAYER=name	Sets the layer name for the entire row. See the LAYER tag for details about layers.
LAYER_OFF	Sets the layer to off (not visible) initially.

```
<TR
    BORDER=number
    BORDERCOLOR=color
    BORDERDASH=number
      [,number,...]
    BORDEROPTS=text
    CORNERSIZE=number
    CORNERSTYLE=text
    BGCOLOR=color
    SHADING=text
    PATTERN=number
    PATTERNCOLOR=color
    3D
    ROWBORDER
    SIDES=text
    LINE_OVER=number
    LINE_UNDER=number
    CELLBACKGROUND=text
    CELLBACKGROUND2=text
    TEXTCLIP=text
    TILE
    TILE2
    TEXTCLIPTILE
    TEXTCLIPREND=number
    CELLSPLIT
    ALIGN=L|R|C|J
    VALIGN=T|M|B
    HEIGHT=number
    FACE=font number
    SIZE=point size
    COMP=number
    COMPTEXT[=OFF]
    DOTTEXT
    NOBREAK[=OFF]
    LINESPACE=number
    CELLXPADDING=number
    FCOLOR=color
    SCOLOR=color
    STRIPEOFF
    STRIPEON
    CONTENTANGLE=number
    CONTENTANGLES=number
    CONTENTBORDERCOLOR=color
    CONTENTBORDERBGCOLOR=color
    CONTENTBORDERPADX=number
    CONTENTBORDERPADY=number
    COLORSPAN_OFF
```

LAYER=name
LAYER_OFF
NO_MULTISORT
FIXHEIGHT=number
MINROWS=number
SUBHEADING=number
[CONTINUE=text]
KEEPCOLSIZE
HIDE>
</TR>

Used to define a table header row. These tags appear inside of the <TABLE> </TABLE> tags. A table row will not split across a page. If there is not enough room for all text in all columns of the row to fit on the current page, a page break will be issued automatically and the table will continue at the top of the next page. All of the parameters are optional. Any values not set default to the values set in the TABLE tag.

Parameter	Description
BORDER=number	Sets the border size. A value of 1 is a line with a width of 1/72 of an inch.
BORDERCOLOR=color	Sets the border color (when the 3D option is not used).
BORDERDASH=number [,number,...]	A number or comma separated list of numbers to specify the ON/OFF length of the border.
BORDEROPTS=text	Used to supply border options in a shorthand format. Specify the border (top, bottom, left or right) followed by a colon then the size, type, and color. For example: "border:1 solid black" is used for a black border on all sides that is a width of 1 point. Use a semi-colon to separate multiple entries such as "border:1 solid black; border-bottom:2 dashed red;". This example will draw a solid 1 point black border on the top, left and right and a red dashed border that is 2 points thick on the bottom. You may use "border:" to set the options for all sides then use "border-top:", "border-bottom:", "border-left:", and "border-right:" to set any of the other borders as needed. Use a width of 0 to turn off a particular side, such as "border:1 solid black;border-top:0;border-bottom:1 dashed green". The type can be either solid, dotted, dashed, or a pair or numbers to represent the on/off length of a dash. The color can be any valid color setting.

Parameter	Description
CORNERSIZE=number	The size of the optional CORNERSTYLE in inches (centimeters if METRIC is used). Consider using CELLPADDING when setting this to a relatively large value so the text remains within the confines of the border.
CORNERSTYLE=text	The optional type of corners to use for the border. The default border is a simple rectangle. Valid values are: Round Bevel Scoop Stair In addition, you may specify a different style for any given corner. Pass a comma separated string containing a code for each corner followed by a colon then the type. The codes for each corner are: TL - Top left TR - Top right BR - Bottom right BL - Bottom left For example, set to "TL:round,TR:scoop,BR:round" to give the top left and bottom right corners a round edge and the top right a scoop edge. The bottom left edge, since it wasn't specified, will have a standard right-angled corner.
BGCOLOR=color	Sets the background color of the row.
SHADING=text	A shading pattern to use for the row. Set to the NAME value from the SHADING tag.
PATTERN=number	A pattern number to use for the background. See the RECT command and its PATTERN option for a sample of each pattern's appearance.
PATTERNCOLOR=color	Color for the pattern.
3D	Sets the border style as 3D which is lighter on top and left and darker on bottom and right. The base color used is the BGCOLOR or BORDERCOLOR if BGCOLOR is not specified.
ROWBORDER	Draws the border around the entire row rather than each cell. Do not use on rows containing cells with the ROWSPAN option.

Tables

Parameter	Description
SIDES=text	Sets what sides to draw the border on. Default is all sides. Values are L (Left), R (Right), T (Top) and B (Bottom). For example, use SIDES=T,B to specify the border should be drawn on top and bottom only.
LINE_OVER=number	Draws a line above each cell in the row. Set number to 1 for a single line or 2 for a double line. In contrast to a border, this line does not extend into the cellpadding around each cell. This option can be used to draw a line for a row of totals for instance.
LINE_UNDER=number	Draws a line below each cell in the row. Set number to 1 for a single line or 2 for a double line. In contrast to a border, this line does not extend into the cellpadding around each cell. This option can be used to underline a heading row for instance.
CELLBACKGROUND=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for all the cells in the row. The image is streched vertically and/or horizontally as needed to fill the area unless TILE is used. This option fills the cell background but not the area included in any cellpadding.
CELLBACKGROUND2=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for all the cells in the row. The image is streched vertically and/or horizontally as needed to fill the area unless TILE2 is used. This option fills the cell background including the area used for cellpadding.
TEXTCLIP=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for the text characters in the row. The image is streched vertically and/or horizontally as needed to fill the cell area unless TEXTCLIPTILE is used.
TILE	Used to tile the background image specified by CELLBACKGROUND.
TILE2	Used to tile the background image specified by CELLBACKGROUND2.
TEXTCLIPTILE	Used to tile the background image specified by TEXTCLIP.
TEXTCLIPREND=number	Rendering mode to use with TEXTCLIP. Range is an integer from 4 to 7. Default is 7.

Tables

Parameter	Description
CELLSPLIT	Allows cells to split across a page boundary. Typically, all cell data in a given row for the outer most table must fit on the current page. A page break occurs and the entire row is moved to the next page if any one cell goes beyond the bottom page margin. This option prints what will fit of each cell on the current page and wraps the remaining text from each cell onto the next page or pages. You may only use this option on rows in the outer most table in the body of the report. This option is for cells containing text only (no embedded tables). Additionally, row options such as MINROWS and ROWSPAN should not be used with CELLSPLIT.
ALIGN=L|R|C|J	Sets the alignment for the cell to Left (default), Center, Right or Justified.
VALIGN=T|M|B	Sets the vertical alignment for the cell to Top (default), Middle or Bottom. Also see the VALIGN2 option. The BASELINE option can be used to set the point to use for middle alignment if you want something different than the middle of the text block.
HEIGHT=number	Forces the row to be at least as high in YUNITS as the number specified.
FACE=font number	Sets the font.
SIZE=point size	Sets the point size for the font.
COMP=number	A percentage to compress the text by. A value less than 100 compresses text while a value greater than 100 expands text.
COMPTEXT[=OFF]	Compress text (when necessary) to fit in a cell rather than perform word wrapping. Use COMPTEXT=OFF to turn option off for the current row.
DOTTEXT	Cuts off the end of any text that doesn't fit the cell and adds three dots (...) to the end of the text portion that does fit.

Tables

Parameter	Description
NOBREAK[=OFF]	Used to prevent page breaks within a set of rows. Add NOBREAK (without the OFF option) on the first row you want to disable page breaks on. On the next row where it doesn't matter where the break occurs, add NOBREAK=OFF to allow a page break on any row. If you have multiple sections all in row where you don't want page breaks, simply issue a NOBREAK on the first row of each section and only put a single NOBREAK=OFF at the end. A NOBREAK section ends at the next NOBREAK or NOBREAK=OFF.
LINESPACE=number	Sets the text line spacing. Each unit is 1/72 of an inch. Default is 2.
CELLXPADDING=number	Sets extra horizontal padding within cells based on units of 1/72 of an inch. You may use decimal values. This is only used when drawing the border.
FCOLOR=color	Sets the fill color. This is also the font color.
SCOLOR=color	Sets the stroke color.
STRIPEOFF	Turns off table row striping. You must use the STRIPE option on the TABLE tag for this to have any effect.
STRIPEON	Turns on table row striping. You must use the STRIPE option on the TABLE tag for this to have any effect.
CONTENTANGLE=number	Pass in the number of degrees (0-360) to rotate cell contents counter-clockwise. Or use a negative number from -360 to 0 to rotate cell contents clockwise. Only the contents of the cell are rotated - not the border or cell background. This sets the default for all cells in the row.
CONTENTANGLES=number	Optional scaling factor to use with CONTENTANGLE. For example, pass in 50 for 50%.
CONTENTBORDER=number	Sets the width in units of 1/72 of an inch of a border drawn around the cell contents. The cell contents is the area of text or graphics drawn inside of a cell. This area can be significantly smaller than the cell size. This border is rotated with the CONTENTANGLE command.
CONTENTBORDERCOLOR=color	The color of the content border.
CONTENTBORDERBGCOLOR=color	The background color of the content border.

Tables

Parameter	Description
CONTENTBORDERPADX=number	Sets extra horizontal padding for the content border based on units of 1/72 of an inch. You may use decimal values.
CONTENTBORDERPADY=number	Sets extra vertical padding for the content border based on units of 1/72 of an inch. You may use decimal values.
COLORSPAN_OFF	Turns off any COLORSTARTx or COLORBGSTARTx tags for the row.
LAYER=name	Sets the layer name for the entire row. See the LAYER tag for details about layers.
LAYER_OFF	Sets the layer to off (not visible) initially.
NO_MULTISORT	Makes the row static and not part of the MULTISORT. See the MULTISORT option on the TABLE tag. For example, use this on a footer row containing totals where you don't want the row to part of the sort - that is, you want the row fixed at the bottom of the table all the time.
FIXHEIGHT=number	Pass a number in points (1/72 of an inch) to force the row to be that height. Note that text will still print beyond the height specified. This option is meant for special situations where you might have a border or image in a row by itself but want to move it up a bit to be flush against the previous row. Use a negative number if necessary to move the row up or a positive number to move it down.
MINROWS=number	Sets the minimum number of rows (including the current one) that must be able to print on the page, otherwise this row is moved to the next page. This is useful when you have data in a column that contains subheading information. In order to ensure that not just the subheading information prints you could set this to a value of 2 or 3. Do not set this to a value that is larger than the number of rows that will fit on a page or you will risk losing information. A value somewhere between 2 and 5 would be reasonable.

Tables

Parameter	Description
SUBHEADING=number [CONTINUE=text]	Labels the row as a subheading row. Subheading rows will reprint on a page break and, unlike a TH row, can be changed anytime during the course of the table. You set SUBHEADING to a number that represents the order it prints when a page break occurs. For example, you may have a table that shows jobs, employees and dates worked by employee for the job. In that case, your TH row would contain Job#, Employee Name and Date Worked. Then, your first TR row could be the actual job# with SUBHEADING=1. The next row would be the first employee for that job# with SUBHEADING=2. You then list each date worked in separate TR rows. When a page break occurs, the job and employee are reprinted just below the heading automatically. Use a negative number to clear out the subheading when you no longer want it to occur. For example, SUBHEADING=-2 will remove the row set with SUBHEADING=2. The CLEARSUBHEADINGS tag may also be used to clear one or more subheadings. Optionally, you can set a CONTINUE message (enclose the text in quotes) to print. You reference the CONTINUE message with the variable &continue in the text between any of the opening and closing TD tags within the TR row. The CONTINUE message only prints when a page break has occurred.
KEEPCOLSIZE	Optional. For use with SUBHEADING and MODTABLE. When creating a SUBHEADING row that will be used after a MODTABLE command, the column sizes from the MODTABLE will be used for all subsequent drawing of the SUBHEADING row. If you wish to retain the column sizes from when the SUBHEADING was first issued, include this option. You may also issue this as a separate tag in the PDF input to apply to all TR tags with a SUBHEADING specified.
HIDE	Prevents the row from printing. This is mostly for the build option using the temporary file. See the Build Options section for more information.

```
<TD
    COLSPAN=number
    ROWSPAN=number
    BORDER=number
    BORDERCOLOR=color
    BORDERDASH=number
      [,number,...]
    BORDEROPTS=text
    CORNERSIZE=number
    CORNERSTYLE=text
    BGCOLOR=color
    SHADING=text
    PATTERN=number
    PATTERNCOLOR=color
    3D
    SIDES=text
    HIDE
    LINE_OVER=number
    LINE_UNDER=number
    CELLBACKGROUND=text
    CELLBACKGROUND2=text
    TEXTCLIP=text
    TILE
    TILE2
    TEXTCLIPTILE
    TEXTCLIPREND=number
    ALIGN=L|R|C|J
    VALIGN=T|M|B
    FACE=font number
    SIZE=point size
    NAME=text
    VALUE=text
    ROTATE=number
    COMP=number
    COMPTEXT[=OFF]
    DOTTEXT
    NOCOMP
    SPAN=TABLE|PAGE
    FILL=text
    FILLLINE
    FILLIMG=text
    LINESPACE=number
    FCOLOR=color
    SCOLOR=color
    CONTENTANGLE=number
    CONTENTANGLES=number
    CONTENTBORDERCOLOR=color
```

CONTENTBORDERBGCOLOR=color
CONTENTBORDERPADX=number
CONTENTBORDERPADY=number
COLORSPAN_OFF
COLORSPANVAL=number
BREAKONCHANGE
LAYER=name
LAYER_OFF
ALTSPAN=number
WIDTH=number
STARTX=number
ENDX=number
OFFSETX=number
NOWRAP
MULTISORTVAL=text
HEADER
PAGE=HARD|SOFT
REPRINT [CONTINUE=text]>
</TD>

Used to define a table data cell. These tags appear inside of the <TH> </TH> and <TR> </TR> tags. All whitespace at the beginning and end of each line of text passed into a TD cell is ignored. Use the &nbsp; command to force spaces when necessary. A table row will not split across a page unless CELLSPLIT is used on the TABLE or TR tag. If there is not enough room for all text to fit on the current page, a page break will be issued automatically and the table will continue at the top of the next page. You may also embed another TABLE inside of this tag. Be careful when embedding tables that you don't insert so much information that the cell doesn't fit on a page. In order for tables to work correctly there must never be more data in a cell than will fit on a single page (except when CELLSPLIT is used). All of the parameters are optional. Any values not set default to the values set in the TH or TR tag.

Parameter	Description
COLSPAN=number	Sets the number of columns to span this cell across. Default is 1.
ROWSPAN=number	Sets the number of rows to span this cell across. Default is 1.
BORDER=number	Sets the border size. A value of 1 is a line with a width of 1/72 of an inch.
BORDERCOLOR=color	Sets the border color (when the 3D option is not used).
BORDERDASH=number [,number,...]	A number or comma separated list of numbers to specify the ON/OFF length of the border.

Parameter	Description
BORDEROPTS=text	Used to supply border options in a shorthand format. Specify the border (top, bottom, left or right) followed by a colon then the size, type, and color. For example: "border:1 solid black" is used for a black border on all sides that is a width of 1 point. Use a semi-colon to separate multiple entries such as "border:1 solid black; border-bottom:2 dashed red;". This example will draw a solid 1 point black border on the top, left and right and a red dashed border that is 2 points thick on the bottom. You may use "border:" to set the options for all sides then use "border-top:", "border-bottom:", "border-left:", and "border-right:" to set any of the other borders as needed. Use a width of 0 to turn off a particular side, such as "border:1 solid black;border-top:0;border-bottom:1 dashed green". The type can be either solid, dotted, dashed, or a pair or numbers to represent the on/off length of a dash. The color can be any valid color setting.
CORNERSIZE=number	The size of the optional CORNERSTYLE in inches (centimeters if METRIC is used). Consider using CELLPADDING when setting this to a relatively large value so the text remains within the confines of the border.
CORNERSTYLE=text	The optional type of corners to use for the border. The default border is a simple rectangle. Valid values are: Round Bevel Scoop Stair In addition, you may specify a different style for any given corner. Pass a comma separated string containing a code for each corner followed by a colon then the type. The codes for each corner are: TL - Top left TR - Top right BR - Bottom right BL - Bottom left For example, set to "TL:round,TR:scoop,BR:round" to give the top left and bottom right corners a round edge and the top right a scoop edge. The bottom left edge, since it wasn't specified, will have a standard right-angled corner.

Parameter	Description
BGCOLOR=color	Sets the background color of the cell.
SHADING=text	A shading pattern to use for the cell. Set to the NAME value from the SHADING tag.
PATTERN=number	A pattern number to use for the background. See the RECT command and its PATTERN option for a sample of each pattern's appearance.
PATTERNCOLOR=color	Color for the pattern.
3D	Sets the border style as 3D which is lighter on top and left and darker on bottom and right. The base color used is the BGCOLOR or BORDERCOLOR if BGCOLOR is not specified.
SIDES=text	Sets what sides to draw the border on. Default is all sides. Values are L (Left), R (Right), T (Top) and B (Bottom). For example, use SIDES=T,B to specify the border should be drawn on top and bottom only.
HIDE	Used to hide the cell on the report. The cell is ignored as far as the PDF is concerened but is included when the data is extracted to Excel using FyTek's PDF to Excel coverter.
LINE_OVER=number	Draws a line above the cell. Set number to 1 for a single line or 2 for a double line. In contrast to a border, this line does not extend into the cellpadding around each cell.
LINE_UNDER=number	Draws a line below the cell. Set number to 1 for a single line or 2 for a double line. In contrast to a border, this line does not extend into the cellpadding around each cell.
CELLBACKGROUND=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for the cell. The image is streched vertically and/or horizontally as needed to fill the area unless TILE is used. This option fills the cell background but not the area included in any cellpadding.

Tables

Parameter	Description
CELLBACKGROUND2=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for the cell. The image is streched vertically and/or horizontally as needed to fill the area unless TILE2 is used. This option fills the cell background including the area used for cellpadding.
TEXTCLIP=text	Set to an image file on disk. Use quotes around the file name. This option uses the image specified as the background for the cell. The image is streched vertically and/or horizontally as needed to fill the cell area unless TEXTCLIPTILE is used.
TILE	Used to tile the background image specified by CELLBACKGROUND.
TILE2	Used to tile the background image specified by CELLBACKGROUND2.
TEXTCLIPTILE	Used to tile the background image specified by TEXTCLIP.
TEXTCLIPREND=number	Rendering mode to use with TEXTCLIP. Range is an integer from 4 to 7. Default is 7.
ALIGN=L|R|C|J	Sets the default alignment for the row - Left (default), Center, Right or Justified.
VALIGN=T|M|B	Sets the vertical alignment for the cell to Top (default), Middle or Bottom. Also see the VALIGN2 option. The BASELINE option can be used to set the point to use for middle alignment if you want something different than the middle of the text block.
FACE=font number	Sets the font.
SIZE=point size	Sets the point size for the font.
NAME=text	Assigns a name to the cell. The name does not have to be unique. You may use the NAME value in functions to add a range of values or perform other calculations.
VALUE=text	An alternate value you wish to use when using a function. For example, you might a cell that displays a graphic or a text string but in calcucations you want to treat this as a particular number. Use this option to specify the numeric or text value to use in function. See the functions section for details on using a function.

Tables

Parameter	Description
ROTATE=number	An angle from 0 to 90 or 270 to 360 to rotate the text in the cell. The default is 0. You must supply the line breaks via the <BR> command if you want the text to be broken into more than one line when using an angle greater than 0. Rotated text is always left aligned.
COMP=number	A percentage to compress the text by. A value less than 100 compresses text while a value greater than 100 expands text.
COMPTEXT[=OFF]	Compress text (when necessary) to fit in a cell rather than perform word wrapping. Use COMPTEXT=OFF to turn option off for the current cell.
DOTTEXT	Cuts off the end of any text that doesn't fit the cell and adds three dots (...) to the end of the text portion that does fit.
NOCOMP	Do not use the computed compression for the table. Instead, force the use of the COMP setting for this cell. If COMP is not used, the default is 100 for 100%.
SPAN=TABLE|PAGE	Spans the cell across the table or page no matter how many columns there are. Useful for section headings in vertically breaking tables where you don't know how many columns to span. This should be the only TD cell in the row when using this feature to avoid overlapping of cells. Setting to TABLE spans just the table and setting to PAGE will span across the entire page (minus margins).
FILL=text	Fills the remaining space between the end of the text and the end of the cell with the character or string specified. Main purpose is to print dots (...) after text in a table of contents to visually line up the text with page numbers or other information. The value for FILL may be any string that will always be used in its entirety. The value will not be split to fit when using more than one character.
FILLLINE	Draws a line to fill the remaining space between the end of the text and the end of the cell.
FILLIMG=text	Fills the remaining space between the end of the text and the end of the cell with the image specified.
LINESPACE=number	Sets the text line spacing. Each unit is 1/72 of an inch. Default is 2.

Parameter	Description
FCOLOR=color	Sets the fill color. This is also the font color.
SCOLOR=color	Sets the stroke color.
CONTENTANGLE=number	Pass in the number of degrees (0-360) to rotate cell contents counter-clockwise. Or use a negative number from -360 to 0 to rotate cell contents clockwise. Only the contents of the cell are rotated - not the border or cell background.
CONTENTANGLES=number	Optional scaling factor to use with CONTENTANGLE. For example, pass in 50 for 50%.
CONTENTBORDER=number	Sets the width in units of 1/72 of an inch of a border drawn around the cell contents. The cell contents is the area of text or graphics drawn inside of a cell. This area can be significantly smaller than the cell size. This border is rotated with the CONTENTANGLE command.
CONTENTBORDERCOLOR=color	The color of the content border.
CONTENTBORDERBGCOLOR=color	The background color of the content border.
CONTENTBORDERPADX=number	Sets extra horizontal padding for the content border based on units of 1/72 of an inch. You may use decimal values.
CONTENTBORDERPADY=number	Sets extra vertical padding for the content border based on units of 1/72 of an inch. You may use decimal values.
COLORSPAN_OFF	Turns off any COLORSTARTx or COLORBGSTARTx tags for the cell.
COLORSPANVAL=number	The cell value to use (when different from the printed cell contents) for COLORSTARTx or COLORBGSTARTx options.

Parameter	Description
BREAKONCHANGE	Set this option on any cell where you want to check if the value has changed. If the value is different from the last time this option was used in the same table in the same column, a page break will occur. You do not need to set this on each row, only the rows where you want to look for a change in a particular cell. For example, suppose you set this on a TD tag with the value ABC in the cell. If you have this set on some later row in the same column as the first TD tag and the value in that cell is XYZ, a page break will be set and the row containing XYZ will start on the new page. The value of XYZ then becomes a new starting value for any later cells with BREAKONCHANGE set.
LAYER=name	Sets the layer name for the entire row. See the LAYER tag for details about layers. Also see the ALTSPAN option in this section.
LAYER_OFF	Sets the layer to off (not visible) initially.
ALTSPAN=number	For use with layers. Use this option to denote this cell as having the specified number of alternate views. For example, if you have the default view and two other views for the cell, specify ALTSPAN=3 on the first occurrence. You would then list two more <TD>...</TD> sets after this cell. All but one of the TD sets should have LAYER_OFF specified (unless you want the text to overlap). Note only the contents of the cell changes - other cells settings such as border or background color should be set the same for all cells in the group.
WIDTH=number	Forces the cell to be at least as wide in XUNITS as the number specified. You may also specify a percentage. Use the FORCEWIDTH option on the TABLE tag if you want to force this value rather than allow the cell width to grow beyond this value if needed.
STARTX=number	Forces the cell for this row only to start at the percentage specified of the total table width. For example, specify 0 to start at the far left up to 100 for the far right. This is useful for times when you want to show a banner or subheading but do not want it to line up with any existing column layout. Do not use a COLSPAN option with this option.

Tables

Parameter	Description
ENDX=number	Forces the cell for this row only to end at the percentage specified of the total table width. For example, 50 to end at the middle of the table. This is useful for times when you want to show a banner or subheading but do not want it to line up with any existing column layout. Do not use a COLSPAN option with this option.
OFFSETX=number	Shifts the position of the cell for this row only by the amount specified in points (1/72 of an inch). Use a negative number to shift the cell to the left or a positive number to shift the cell to the right. Do not use a COLSPAN option with this option.
NOWRAP	Forces the cell to remain wide enough to fit the requested text without wrapping. For a wide table this is achieved by compressing the text in the table so you should be careful not place too much text in a cell with NOWRAP specified.
MULTISORTVAL=text	The value to use for the cell contents for sorting purposes when using the MULTISORT table option. The default is to sort based on the cell contents when this option is not used. You may want to set this to a different value for sorting. For example, to include a secondary field in the sort that isn't part of the sort column. You may have a name in one column for a sort but you want to sub-sort based on some other id when the names are the same. You'd set the full string to sort on with this option, for example, "Smith,Bob,0023" where the cell contents between the TD and /TD tags is "Bob Smith". The MULTISORTVAL is not displayed anywhere as part of the PDF output.
HEADER	Specifies the column as a header column which will reprint on page breaks for tables that break vertically (see the BREAK option on the TABLE tag). You only need to specify this on a single TR/TH row anywhere in the table.

Parameter	Description
PAGE=HARD|SOFT	Specifies where a vertical page break should occur in the table (see the BREAK option on the TABLE tag). You only need to specify this on a single TR/TH row anywhere in the table. Setting this option to HARD will force a vertical page break on the specified column. Setting to SOFT will imply that the column is a candidate for a vertical page break. If any column has a setting of SOFT then those columns without the PAGE option will be kept together and not broken.
REPRINT [CONTINUE=text]	Used to reprint the contents of a cell after a page break. Optionally, you can set a CONTINUE message (enclose the text in quotes) to print. You reference the CONTINUE message with the variable &continue in the text between the opening and closing TD tag. The CONTINUE message only prints when a page break has occurred. For example, you may have a column in your table where the value remains the same over many rows. Rather than print the value every time, you can set REPRINT on the first occurrence of the value and leave the cell blank for all rows following until the value changes. If a page break occurs the REPRINT option forces the value to be printed in the first row after the heading.

<TABLEPAGE
YPOS=number>

Used to force a page break while inside of a table. Only issue this command between rows. That is, after a closing TR and before the TR whose row you want to start on the next page. This command serves as a shortcut to closing the table, issuing a PAGE command then opening another table with the same layout. All table headers will be repeated just like when a standard page break occurs.

Parameter	Description
YPOS=number	Conditionally page break based upon the current Y position. For example, if the current GRID settings are 100 in the Y direction and you set YPOS=70, the TABLEPAGE command will be ignored if the current Y position is less than 70. You may also enter the value followed by a % sign (i.e. YPOS=65%) to denote a percentage of the current GRID.

<CLEARSUBHEADINGS
VALUE=text>

Used to clear subheadings set by previous table rows with the SUBHEADING option. You may also clear subheadings by specifying a negative value with a SUBHEADING option on the next table row. However, this command allows you to clear multiple subheadings at once without a table row. Only issue this command between rows.

Parameter	Description
VALUE=text	A comma separated list of one or more subheading numbers to clear. For example, "2,3,5" will clear subheadings 2, 3, and 5.

SQL Queries

SQL Queries are used to extract information from a data source using SQL commands. The SQL statements can be included in the input for Report Writer to process, loop through and dynamically generate a report. In addition, you may use conditional statements within the report to format as needed based on the results of the query. The -allowperl option or setAllowPerl method must be specified when executing Report Writer for the conditional statements to be processed. You may need to use the option -perldbi on Unix systems for processing CSV files. Use -sqlcmds on Unix to load the needed SQL library. Because of differences among Unix systems, the libraries may not be compatible so they are not loaded unless this option is used.

Pound signs are used to enclose column names from queries when you want to display the data from the query. For example, if you have a column named "city" in a query named "customers" you would use #customer.city# in the commands passed to Report Writer. The string #customers.city# will be replaced with the data for the current record. The sytax in the input file for Report Writer would look something like <TD>#customers.city#</TD>. In calculations or commands like RWGET, use $customers.city instead. The reason for this is to prevent issues with embedded quotes. With the above example, #customers.city# is the same as the tag <RWGET $customers.city>. Both will return the same value. In a calculation, using <RWIF #customers.city# ne "">, the program will perform the replacement first and you may have something like <RWIF Los Angeles ne ""> which will not work as there are no quotes around Los Angeles. When using <RWIF $customers.city ne ""> instead, the program will use a variable with the value equal to Los Angeles during the evaluation process and work as expected.

You may define functions within your code and perform if/then/else processing as well. All other interaction with data from queries and in functions use Perl syntax. Perl has three data types you should be aware of. The first is a scalar which is used for numbers or strings. Variables of this type will all start with a $ as in $myvar. The second is an array type. Variables of this type will all start with a @ such as @myarray. The last is a hash type and those start with a %. It is beyond the scope of this documentation to cover programming in Perl. If you are at all familiar with JavaScript then much of the syntax will look similar.

The sample program sample_sql.frw that comes bundled with Report Writer has many examples that may be helpful. Be sure to specify the option -allowperl or the method setAllowPerl when you run that example. Some common operators are:

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
$a = $b	Assignment
$a == $b	Check if two numbers are equal
$a != $b	Check if two numbers are not equal
$a eq $b	Check if two strings are equal
$a ne $b	Check if two strings are not equal
$a lt $b	Check if a is less than b (in Perl, you can use < but since < also starts a tag, lt should be used instead)
$a gt $b	Check if a is greater than b
$a le $b	Check if a is less than or equal to b
$a ge $b	Check if a is greater than or equal to b
cond1 || cond2	Logical OR
cond1 && cond2	Logical AND

The LOOP tag is the indicator to Report Writer that the following content up until the closing LOOP tag will contain functions for evaluation. For example, if you place the text <RWGET $abc> in a TEXT block, the tag itself will be displayed. When you wrap the TEXT block in a LOOP block, any commands such as this are evaluated before the TEXT block processes it. You may place RWSET and RWGET outside of any text and loop blocks and they will processed. Only when you have commands inside of TEXT blocks or TD table cells you want to evaluate and you are not already inside of an outer loop block do you need to wrap these inside an outer LOOP block.

Use the -debug and/or -e command line options (setDebug and/or setErrFile DLL methods) to report on Perl errors. The debug output will show the contents of what was passed in to Report Writer along with any queries it processed and the time to process along with the number of rows returned.

Database Connection

The database connection, specified with the -sqldb or setSQLDB method, is used to set the parameters necessary for connecting to your database. Each type of connection has different options. In addition, you may specify the connection options on the QUERY tag. The -sqldriver option should be set to one of the below drivers (such as "Oracle"). Based on the driver below, determine what you should pass to -sqldb based on the description.

CSV/CSVPP/Fixed/Tab/Pipe
Specify the directory where the files are located with the syntax "f_dir=directory" where "directory" is the path to the files. To specify the current directory you would use "f_dir=." Use "csv_sep_char=" to specify the field separator when other than a comma. For example, "f_dir=.;csv_sep_char=|" will set the pipe character as the field separator. Assign your tables using the RWTABLE tag.

Oracle (32-bit only)
Pass just the database or leave the option off and pass as part of the user such as "username@XE". Or, you may pass the host, SID and port as in "host=localhost;sid=XE;port=1521". Oracle version 8 through 11g should be able to connect.

mysql (or mysqlPP)
Specify the database name and the host IP address. For example, "database=mydb;host=localhost". If necessary, you may specify the port as well such as "database=mydb;host=localhost;port=3306".

ODBC
Specify the DSN you wish to connect.

XML
You do not need to pass anything when using XML. Assign your tables using the RWTABLE tag.

```
<RWTABLE
    FILE=text
    TABLE=text
    FORMAT=text
    FLAGS=text
    COLNAMES=text
    EOL=text>
```

Only used when using CSV or XML files as the data source. You do not need this for other databases such as Oracle or MySQL.

Used to define a mapping between files on disk and logical table names in queries. For example, you might want to map a file called "employees.csv" to the table "empl". This allows you to then refer to the table "empl" in your queries. Add one of these tags for each table you will need.

Parameter	Description
FILE=text	The name of the physical file on disk. This may also be a web page that starts with http://.
TABLE=text	The logical name for this data to use in the QUERY tag. This is the name to use as the name of the table.
FORMAT=text	Set to one of the following: CSV XML Fixed Pipe Tab
FLAGS=text	Use to specify the root node for XML or columns widths for fixed format. For XML, specify by using the string record_type=>'node'. For example, if the nodes are called item that you are interested in looping through, set FLAGS="record_type=>'item'". For fixed width format (FORMAT=Fixed), specify the widths of the fields and set COLNAMES. For example, if you have a field 20 characters long followed by a field 5 characters long you would set FLAGS="pattern=>'A20 A5'".

Parameter	Description
COLNAMES=text	Optional. Only include if the file you are using does not contain field names in the first row of the file or you are using a fixed width file. Set to a space separated list of column names you want to use to refer to each column in the file. For example, COLNAMES="name addr1 addr2 city state zip".
EOL=text	Optional. The End Of Line separator for CSV files. Set to \r\n for DOS files or \n for Unix formatted files.

```
<QUERY
    NAME=text
    DSN=text
    PROCEDURE=text
    SQLDRIVER=text
    SQLDB=text
    USERID=text
    PASSWORD=text>
</QUERY>
```

Used to define a query. Place your SQL select statement between the opening and closing QUERY tags. Only the NAME is required. The connection options may be passed on the command line or via DLL methods. You may enter connection options on the query tag if you want to use different settings.

Use the PROCEDURE option to provide the procedure name when calling a stored procedure. In this case, rather than SQL, you place QPARAM tags between the opening and closing QUERY tags. Each QPARAM tag corresponds to a variable passed in or out of the procedure.

Use a select statement to return a value from a database function. For example, if you have an Oracle function named myfunct that takes two parameters as input you would write the select like this:

```
<QUERY NAME="runfunct">
select myfunct(<QPARAM $var1>, <QPARAM $var2>) res from dual
</QUERY>
```

You can then refer to the result with the variable $runfunct.res in your code.

Parameter	Description
NAME=text	Provide a unique name for the query. The name is used in the LOOP statement to iterate over the result set.
DSN=text	The DSN name from the Report Writer DSN configuration file to use. DSN names are setup when running Report Writer in server mode and using the -dsnfile option. You do not need to set SQLDRIVER, SQLDB, USERID, or PASSWORD in this case.

Parameter	Description
PROCEDURE=text	Only specify this if you are calling a stored procedure. Provide the name of the procedure only. The parameters will be passed using the QPARAM tags before the closing </QUERY> tag. For example, if you have a stored procedure in Oracle named VALID_USER(ID NUMBER, LEVL OUT VARCHAR2) you would set PROCEDURE="VALID_USER" (or PROCEDURE="SCHEMA.VALID_USER" where SCHEMA is the schema for the procedure). You would then provide two QPARAM tags - the first for ID (the input) and the second of LEVL (the output).
SQLDRIVER=text	The data source. This is a case-sensitive string. Not necessary when using the DSN option. Available for Windows/Linux only. Valid values are: CSV (or CSVPP) Fixed Tab Pipe Oracle (32-bit only) mysql (or mysqlPP) ODBC XML CSV may give slightly better performance over CSVPP on Windows systems. CSVPP, Fixed, Tab, and Pipe do not support all SQL options.
SQLDB=text	The database schema or driver information. Not necessary when using the DSN option. See the Database Connection section for details.
USERID=text	The user id (if any) for the database connection. Not necessary when using the DSN option. For Oracle, you may also specify the password and/or schema in this field. For example, "user/pwd" or "user/pwd@prod".
PASSWORD=text	The password (if any) for the database connection. Not necessary when using the DSN option.

<QPARAM variable>
-- or --
<QPARAM
TYPE=text
VAR=text
VALUE=text>
</QPARAM>

Used to prevent SQL injection by passing parameters to the query during execution rather than placing directly in the SQL statement. The "variable" is the variable you want to use, either $var (from Perl) or $query.column (from a prior query) syntax. This option is also used to pass parameters with stored procedures.

For example, when you do this:

```
<QUERY NAME="getCities">
select city, state, zip from cities
  where company = '<RWGET $comp>'
</QUERY>
```

The result that is parsed is this which is not valid because of the quotes:

```
<QUERY NAME="getCities">
select city, state, zip from cities
  where company = 'O'Reilly'
</QUERY>
```

When you use QPARM like this:

```
<QUERY NAME="getCities">
select city, state, zip from cities
  where company = <QPARAM $comp>
</QUERY>
```

The result is now this with a placeholder where the value will go:

```
<QUERY NAME="getCities">
select city, state, zip from cities
  where company = ?
</QUERY>
```

The ? is used as a placeholder that is evaluated later during processing. It is replaced by the string "O'Reilly" when needed. This also protects against unwanted commands being passed if your variables are based on user input.

For stored procedures, there are several other options for QPARAM. For example, assume you have a stored procedure in Oracle setup as VALID_USER(ID NUMBER, LEVL OUT VARCHAR2). You would then provide two QPARAM tags for the parameters. For example: The result that is parsed is this which is not valid because of the quotes:

```
<QUERY NAME="chkUser" PROCEDURE="VALID_USER">
```

```
  <QPARAM TYPE="IN" VAR="p_id" VALUE="100">
  <QPARAM TYPE="OUT" VAR="p_level" VALUE="$plevel">
</QUERY>
```

The value for ID is set to 100 in this example. This could also be a Perl value that was set from a prior RWSET or a value from a prior query if you like. In that case, you would have a variable such as $pid or $myquery.column. The value output from the query will be placed in the Perl variable $plevel. You can then display it or use $plevel in other functions as if you defined it with an RWSET statement.

Parameter	Description
TYPE=text	Set to IN, OUT or INOUT. This is only used for stored procedure. Represents the type of parameter this is for the procedure.
VAR=text	The variable placeholder name for the stored procedure. This is simply used to name the placeholder so it does not have to correspond to any prior variable. Each VAR should be unique though within the QUERY section.
VALUE=text	The value to pass in specified as either a hard-coded value or a Perl variable that has been initialized. Or, this is the name of the variable to receive the output of the stored procedure. Output or input-output parameters should always specify a Perl variable in the form $name where "name" is the name of your variable.

<LOOP
QUERY=text
DUMP
FROM=number
TO=number
-- or ---
INDEX=text
ARRAY=text
SORTSTR[=text]
SORTNUM[=text]
SORTDESC
-- or ---
INDEX=text
LIST=text
DELIMITER=text
-- or ---
INDEX=text
FILE=text>
</LOOP>

Used to loop over the result set for the named query, an array, a list, or a file. Place any Report Writer syntax between the opening and closing LOOP tags. Use the # symbol to enclose variables you want replaced by the query results.

For example, if you query looks like this:

```
<QUERY NAME="getCities">
select city, state, zip from cities
</QUERY>
```

You can use #getCities.city#, #getCities.state#, or #getCities.zip# as variables for display between the opening and closing LOOP tags. Be sure to use the name of the query followed by a dot then the field name.

In addition, you can use $getCities.city in any Perl code or QPARAM statements. The query.column sytanx used in any Perl code will be converted to a variable holding the field value.

You can also nest QUERY and LOOP commands inside of a loop. This allows you base lower level queries on an upper level one. You may use variables in your SQL when inside a loop to create a dynamic query. For example:

```
<QUERY NAME="getCities">
select city, state, zip from cities
</QUERY>
<LOOP QUERY="getCities">
  ...RW commands (can use #getCities# here)...
  <QUERY NAME="getBusinesses">
  select business_name, business_type from businesses
    where city = <QPARAM $getCities.city>
    order by business_name
  </QUERY>
  <LOOP QUERY="getBusinesses">
    ...RW commands (can use #getCities#
            or #getBusinesses# here)...
  </LOOP>
  ...RW commands (can use #getCities# here)...
</LOOP>
```

If you just need to output a variable and don't need to loop over anything, use the LOOP command without any options to display the value or perform command processing. For example, if you just want to print the value of a variable passed in you may do this:

```
<LOOP>
<TEXT>
  inpvar = <RWGET defined($inpvar)>
</TEXT>
</LOOP>
```

Note the LOOP command goes outside of the TEXT or TD tags. If you are already inside an outer loop, there is no need to add an inner loop just for the variable. Without the LOOP command above, the text would display the RWGET command rather than process it.

Parameter	Description
QUERY=text	This is the only mandatory option when looping through a query result set. Provide the name of a previously defined QUERY statement you wish to loop through.
DUMP	Provides a quick and dirty dump of the data. You might use this in the initial setup phase to get the list of column names and some sample data from your data source. This option will automatically create a table whose column headings are the column names from the query. Use the FROM/TO option to limit your results to a small set. In general, you will have the opening LOOP tag followed by the closing LOOP tag when using this option. You do not need an opening TABLE or TEXT command above the LOOP statement since this will generate a TABLE statement for you.

Parameter	Description
FROM=number	The starting row number (default is the first). Use a negative number to mean an offset from the total number of records. For example, -10 means the starting row the total number of rows - 9, or 10 from the end.
TO=number	The ending row number (default is the last). Use a negative number to mean an offset from the total number of records. For example, -1 means the last row. Using FROM=-3 and TO=-1 would select the last 3 rows of the query.
INDEX=text	The variable name when looping through an array, list, or file. Use a $ in front of the variable name. For example, $i.
ARRAY=text	The array to loop through. This may be a simple array or an array of hashes. The LOOP command would be: <RWSET @ary = ("apple","orange","banana")> <LOOP INDEX="$idx" ARRAY=@ary>
SORTSTR[=text]	Sort the array as a list of strings. Optionally, if you have an array of hashes, specify the hash column.
SORTNUM[=text]	Sort the array numerically. Optionally, if you have an array of hashes, specify the hash column.
SORTDESC	Sort in decending rather than the default of ascending.
LIST=text	The list to loop through. A list can be a variable or static list. For example, LIST="a,b,c" or LIST="$mylist". The LOOP command would be: <LOOP INDEX="$idx" LIST="a,b,c">
DELIMITER=text	The delimiter to use for the text. The default is a comma.
FILE=text	The file to loop through. Each line of the file is one iteration of the loop. The LOOP command would be: <LOOP INDEX="$idx" FILE="c:\temp\myfile.dat">

<RWSET variable=text>

Used to assign some value to a variable. The variable can be a string, array, or hash. For example, $i = 1 will set the value of $i to 1. Assuming we have a query named customers that contains a column called city, then $c = $customers.city will set $c to the current value of city. The current value is based on what row we happen to be on. In this case, the RWSET statement would be inside of the LOOP that iterates over the query result set. You may use any valid Perl assignments or short-hand notation such as $i += 1 which sets $i to itself plus 1.

<RWFUNCT
NAME=text
</RWFUNCT>

This is used with the functions feature to store a commonly used function to a variable. For example:

```
<RWFUNCT NAME='calc'>
numformat(fn(name:'qty',thisrow,isnum) *
fn(name:'price',thisrow,isnum),'$##,###.##')
</RWFUNCT>
```

The value for calc may then be referenced with <RWGET funct(calc)> anywhere the function is to be evaluated.

<RWGET text>

Retrieves a variable for display which may be a Perl variable or a query column value. In addition, you may use this tag to perform other functions before displaying the value or for formatting. For example, <RWGET $c> will retrieve the value for $c. Use the notation $query.column when referring to a column from a query rather than #query.column#. You may also use <RWGET add_tax($customer.payment)>. In this case, it is assumed you have a Perl function defined in the input called add_tax. This function will take the value of payment from the customer query and perform some type of formatting. The result is then displayed in the report.

Use the function numformat(x, mask [,dec, thousands sep]) to format numbers. In this case, x is the number to format and mask is the format mask. Optionally include the decimal character and thousands separator which would be the comma and period for European formats. Use the # symbol for numbers in the mask and be sure to make the mask large enough for the biggest number that might be returned. For example, numformat(1234.5678,'#,###.##') would result in 1,234.57 being displayed.

You may also apply functions in RWGET for use in column calculations within a table. This is for situations outside of a query loop only. In this case, use the function named fn and pass parameters for what columns and rows to apply the function to.

You do not need to use RWGET in other statements such as RWSET. This statement is only a wrapper for applying additional formatting before display or to display user defined variables set with RWSET.

```
<RWIF condition>
    ...statements...
    [<RWELSEIF condition>
    ...statements...]
    [<RWELSE>
    ...statements...]
</RWIF>
```

This statement is used to provide conditional processing. Any valid Perl constructs (with some additions below) may be used to create the condition to check.

Operator	Description
a lt b	String compare if a < b
a gt b	String compare if a > b
a le b	String compare if a <= b
a ge b	String compare if a >= b
a eq b	String compare if a = b
a ne b	String compare if a <> b
a nlt b	Numeric compare if a < b
a ngt b	Numeric compare if a > b
a nle b	Numeric compare if a <= b
a nge b	Numeric compare if a >= b
a == b	Numeric compare if a = b
a != b	Numeric compare if a <> b

These operators are used instead of the < or > symbols so Report Writer does not match the < or > symbols as the start or end of a tag. You may nest other RWIF statements as part of the ...statements... above. Also, you may include all the other commands such as RWGET/RWSET as well as LOOP and QUERY. This allows you to conditionally parse other queries and loop through the results.

This example shows what you might put in a table cell based on the current customer's balance. This code would go between the <TD> and </TD> tags in a table or between <TEXT> and </TEXT> tags.

```
<RWIF $customer.balance ngt 0>
  <QUERY NAME="inv">
    select invoice, due_date from invoices
      where id = '#customer.id#'
  </QUERY>
  <LOOP QUERY="inv">
    Inv#: #inv.invoice# Due: #due_date#<br>
  </LOOP>
<RELSEIF $customer.balance nlt 0>
  Refund is due to customer.
<RELSE>
  Customer does not have a balance.
</RWIF>
```

<RWSCRIPT>
</RWSCRIPT>

Used to define Perl values or sub-routines that can be called with the RWGET tag. Place any valid Perl syntax between the opening and closing RWSCRIPT tags.

The function names are case-sensitive. For example, this scripts places commas in a number to format it for display.

```
<RWSCRIPT>
  sub addcommas {
    my $input = shift;
    $input = reverse $input;
    $input =~ s<(\d\d\d)(?=\d)(?!\d*\.)><$1,>g;
    return reverse $input;
  }
</RWSCRIPT>
```

RecordNum

"recordnum" is a built-in function that returns the current row number you are on while looping through a query. Specify the query name followed by a period then the text recordnum. It is treated as a column name that is available in all queries. For example:

```
<LOOP QUERY="cities">
  Recnum = #cities.recordnum#<BR>
  Recnum = <RWGET $cities.recordnum><BR>
</LOOP>
```

RecordCount

"recordcount" is a built-in function that returns the total number of rows matched by a query. Specify the query name followed by a period then the text recordcount. It is treated as a column name that is available in all queries. For example:

```
Recnum = #cities.recordcount#<BR>
Recnum = <RWGET $cities.recordcount><BR>
```

first(text)

"first" is a built-in function to determine if you are on the first iteration of your query. Provide the query and column you want to check in the form query.column. If you are at the first record then the result is true. Pass in the column you want to check.

For example, <RWIF first('customer.state')> would result in true if this is the first record in the loop. You should sort your results by using an order clause in a way that this function can use it. In this case, we would be sorting customers by state. As we loop through the query, the first iteration of the loop will result a value of true.

firstof(text)

"firstof" is a built-in function to determine if you are on the first iteration of a particular value in your query. Provide the query and column you want to check in the form query.column. If you are at the first record or the prior record had a different value for the field then the result is true. Pass in the column you want to check.

For example, <RWIF firstof('customer.state')> would result in true if this state is the first for it's value. You should sort your results by using an order clause in a way that this function can use it. In this case, we would be sorting customers by state. As we loop through the query, the first time we hit any given state this results to a value of true. If we have 20 entries for California, the first entry we come across for CA will result in this being true. The next 19 entries for CA will result in this result being false.

last(text)

"last" is a built-in function to determine if you are on the last iteration of your query. Provide the query and column you want to check in the form query.column. If you are at the last record then the result is true. Pass in the column you want to check.

For example, <RWIF last('customer.state')> would result in true if this is the last record in the loop. You should sort your results by using an order clause in a way that this function can use it. In this case, we would be sorting customers by state. As we loop through the query, the last iteration of the loop will result a value of true.

lastof(text)

"lastof" is a built-in function to determine if you are on the last iteration of a particular value in your query. Provide the query and column you want to check in the form query.column. If you are at the last record or the prior record had a different value for the field then the result is true. Pass in the column you want to check.

For example, <RWIF lastof('customer.state')> would result in true if this state is the last for it's value. You should sort your results by using an order clause in a way that this function can use it. In this case, we would be sorting customers by state. As we loop through the query, the last time we hit any given state this results to a value of true. If we have 20 entries for California, the last entry we come across for CA will result in this being true. The previous 19 entries for CA will result in this result being false.

removespaces(text)

"removespaces" is a built-in function to remove extra spaces and line-feeds from text. Use this when you have XML data, for example, that spans several lines in the XML file. This function will remove spaces from the front and back of the text as well as collapse any multiple spaces within the text down to one. For example:

```
<LOOP QUERY="books">
  <RWGET removespaces($books.description)><BR>
</LOOP>
```

JSON Data Merge

<JSON
SRC=text
NAME=text
FROM=number
THRU=number>
</JSON>

The JSON (JavaScript Object Notation) tag is used to import JSON formatted data from a file and insert into a TABLE or other structure. Place your template (Report Writer syntax) between the opening and closing JSON tags. JSON tags may not be placed inside a TD or TEXT block as they will not be processed. The name variables from your JSON name/value pairs will typically be placed within TD or TEXT blocks for display.

You may embed an opening and closing JSON block within an outer JSON tag set to loop through an array or simply denote the outer node you wish to work with. The outermost JSON tag must have the SRC option specified which is the file containing your JSON data. Any inner JSON tags must have the NAME option specified which is the node you want to refer to within the template.

A $ is used to denote a JSON variable name in the Report Writer syntax. This may be a simple variable such as $myvar or a variable within a node such as $myvar.rpt.title. You may also refer to an array element such as $myvar[n] or $myvar.rpt[n].title where n is the 0 based array entry.

Parameter	Description
SRC=text	The path and file name containing the JSON data. Set this once then embed additional JSON tags (if necessary) with just the NAME option to traverse those nodes.
NAME=text	The name of the node to access relative to the current level. For example, if you have a structure that has a node "table" with nodes "title" and "color" you may specify "table" for the NAME parameter then refer to $title and $color between this tag and the corresponding JSON tag. The contents between the opening and closing JSON tag are repeated once for each entry of the array when NAME evaluates to an array.
FROM=number	Optional. The starting index (0 based) to use when the NAME option refers to an array. Default is 0.

Parameter	Description
THRU=number	Optional. The ending index (0 based) to use when the NAME option refers to an array. Default is the number of elements in the array minus one.

The $<var> notation is used between the opening and closing JSON tags to render the values. For example, in your input file you might have a table cell that you want to display the value of a field called "mydata" within a "table" array. In the input file for Report Writer you would specify that as:

```
<PDF>
<PAGE>
<TABLE>
<JSON SRC="myjson.txt" NAME="table">
  <TR>
    <TD>$mydata</TD>
  </TR>
</JSON>
</TABLE>
```

The JSON file (named myjson.txt based on this example) could look like this:

```
{
  "table" : [{"mydata": "line 1"},
             {"mydata": "line 2"},
             {"mydata": "line 3"}
             ]
}
```

The PDF output would be a table consisting of 3 rows with line 1, line 2 and line 3 in the table rows.

You may also use the dot notation to refer to values. For example, you could leave the NAME option off the JSON tag in the above example and use $table.mydata in the table cell instead. Any node that is an array will repeat the report syntax between the opening and closing JSON tag for each array entry.

Here is another slightly more complex example.

```
<pdf>
<page>
<json src="myfile.json">
  <json name="tbl">
    <table size=10 border=.5>
      <th bgcolor='$th.bgcolor'>
      <td>$th.col1</td>
      <td>$th.col2</td>
      <td>$th.col3</td>
      </th>
      <json name="td" thru=1>
        <tr><td>$col1</td>
        <td>$col2</td>
        <td>$col3</td></tr>
    </json>
  </json>
  </table>
  <text>
    $notes
    <br>
    $tbl.td[1].col2
  </text>
</json>
```

The JSON file (named myjson.json) for the above example:

```
{
  "tbl": {
    "th": {
      "bgcolor" : "#99ccff",
      "col1" : "Col-1",
      "col2" : "Col-2",
      "col3" : "Col-3"
    },
    "td" :[{"col1" : "1-A", "col2": "1-B", "col3": "1-C"},
           {"col1" : "2-A", "col2": "2-B", "col3": "2-C"},
           {"col1" : "3-A", "col2": "3-B", "col3": "3-C"}
           ]
    },
    "notes": "Test text"
}
```

Charts

Charts are used to graphically display numeric information. There are several types of charts including BAR/LINE, AREA, RIBBON, PIE, SCATTER or STOCK. The type of data you wish to display will determine which chart works best. Additionally you may also use a graph. A graph differs from a chart in that a graph is an image (jpeg) instead of a vector based chart drawing within the PDF. The graph feature uses the Perl GD::Graph library to render the graph. See the Graphs section for more details.

Charts are placed outside of any TEXT block or within a TABLE cell. Leave extra padding around a chart in a table cell as it's possible the chart may go beyond the cell boundary, depending on the chart legend or axis labels.

Charting functions are not available with PDF Report Writer SE.

```
<CHART
     WIDTH=number
     HEIGHT=number
     X1=number
     Y1=number
     X2=number
     Y2=number>
</CHART>
```

Used to define a chart.

Parameter	Description
WIDTH=number	Sets the width of the chart in units based on the GRID command. You may display a chart in a table cell in which case you may just enter the width and height rather than X and Y positioning. Chart commands must be issued outside of TEXT blocks or within a TD tag.
HEIGHT=number	Sets the height of the chart in units based on the GRID command.
X1=number	Sets the value of the left edge in units based on the GRID command. Default is 0.
Y1=number	Sets the value of the top edge of the chart in units based on the GRID command. Default is the current position.
X2=number	Sets the value of the right edge in units based on the GRID command. Default is the value of XUNITS from the GRID command.
Y2=number	Sets the value of the bottom edge in units based on the GRID command. Default is the value of YUNITS from the GRID command.

Within the <CHART> and </CHART> tags go the parameters described next. Once the </CHART> tag is processed the chart is rendered.

```
<PARAMS
    TYPE=text
    3D=number
    3DBOX
    3DTOP=number
    3DSIDE=number
    3DCOLORTOP=color
    3DCOLORSIDE=color
    3DLEFTCOLOR=color
    3DLEFTSHADE=text
    3DBOTTOMCOLOR=color
    3DBOTTOMSHADE=text
    3DBACKCOLOR=color
    3DBACKSHADE=text
    BORDEROPTS=text
    ID=text
    HORIZONTAL
    BEZIER
    LEFTDATALABEL=number
    BGCOLOR=color
    TRANSPARENCY=name
    BARSTYLE=text
    BARTRANSPARENCY=name
    BARDATA=list
    HIGH=list
    AVG=list
    LOW=list
    STOCKSHADE=color
    DATALAB=list
    DATALABBGCOLOR=color
    LABFONT=number
    LABSIZE=number
    LABCOLOR=color
    LABCOMP=number
    BARSPACE=number
    BARMARGIN=number
    BARMAX=number
    PIEOFFSET=number
    BARCOLOR=list
    BARSHADING=list
    BARSHADINGTOP
    REVEFFECT
    ALLBARCOLOR=number
    ALLBARSHADING=text
    BARPATTERN=list
    IMAGE=text
    IMAGETOP=text
```

IMAGETOPVALIGN=text
IMAGEBOT=text
IMAGETOPREV=text
IMAGEBOTREV=text
LINECOLOR=list
LINENODE
NODEWIDTH=number
LINELAB
COMPACTLEGEND
LEGEND=list
LEGENDBGCOLOR=color
LEGENDSHADE=text
LEGENDOFFSET=number
LEGENDLABWIDTH=number
LEGENDCOLS=number
LEGENDPCT=number
LGLINESPACE=number
NODESIZE=number
NODE=number
NODEFCOLOR=color
NODESCOLOR=color
BASEFCOLOR=color
BASESCOLOR=color
XGRID=number
YGRID=number
GRIDON=number
GRIDOFF=number
GRIDDASH=number,number[,number,number...]
GRIDFCOLOR=color
GRIDSCOLOR=color
SHOWXTICKS=On|Off
XSCALE=list
YSCALE=list
YSCALE2=list
YSCALEPFX=text
YSCALESFX=text
DATAMULT=number
DATAMLAB=list
VALUETOP
BARCOLUMNS
LEGENDCOLOR=list
NOCOLORLEGEND
MULTCOLORLEGEND
OUTSIDELABELS
XSCALEFONT=number
XSCALESIZE=number
XSCALECOLOR=color

XSCALECOMP=number
YSCALEFONT=number
YSCALESIZE=number
YSCALECOLOR=color
YSCALECOMP=number
XAXISLABEL=text
XAXISFONT=number
XAXISSIZE=number
XAXISCOLOR=color
XAXISCOMP=number
XAXISOFFSET=number
YAXISLABEL=text
YAXISLABEL2=text
YAXISFONT=number
YAXISSIZE=number
YAXISCOLOR=color
YAXISCOMP=number
YAXISOFFSET=number
YAXISOFFSET2=number
XLABEL=list
XLABELFONT=number
XLABELSIZE=number
XLABELCOLOR=color
XLABELCOMP=number
YLABEL=list
YLABEL2=list
XLABELOFFSET=number
YLABELOFFSET=number
XLABELOFFSET2=number
YLABELOFFSET2=number
XLABELPOS=number
YLABELPOS=number
TOPLABELS
LABANGLE=number
FONT=number
SIZE=number
COLOR=color
COMP=number
TITLE=text
TITLEFONT=number
TITLESIZE=number
TITLECOLOR=color
TITLECOMP=number
TITLEOFFSET=number

Used to define parameters for the chart. You may issue multiple PARAMS tags within the CHART tag so you don't need to stuff all of the parameters into one long PARAMS tag.

Charts

Parameter	Description
TYPE=text	One of the following: BAR-LINE, AREA, RIBBON, PIE, SCATTER, STOCK or RADAR. Not all parameters are valid for all chart types.
3D=number	Sets the depth of the chart based on units of 1/72 of an inch. Usually a value between 5 and 20 is best.
3DBOX	Outlines the front of 3D charts.
3DTOP=number	The amount to lighten or darken the top of 3D bars. Set to a value between -100 and 100. Negative values will darken and positive values will lighten. The color used is based on the color of the bar. The default is -25.
3DSIDE=number	The amount to lighten or darken the side of 3D bars or the side of 3D pie charts. Set to a value between -100 and 100. Negative values will darken and positive values will lighten. The color used is based on the color of the bar. The default is -17.5.
3DCOLORTOP=color	Optional. Color to use for the top of 3D bars. Any valid RGB color may be used (no CMYK or Pantone colors).
3DCOLORSIDE=color	Optional. Color to use for the side of 3D bars. Any valid RGB color may be used (no CMYK or Pantone colors).
3DLEFTCOLOR=color	Background color for the left pane of a 3D chart. Any valid RGB color may be used (no CMYK or Pantone colors).
3DLEFTSHADE=text	Background shading pattern for the left pane of a 3D chart.
3DBOTTOMCOLOR=color	Background color for the bottom pane of a 3D chart. Any valid RGB color may be used (no CMYK or Pantone colors).

Parameter	Description
3DBOTTOMSHADE=text	Background shading pattern for the bottom pane of a 3D chart.
3DBACKCOLOR=color	Background color for the back pane of a 3D chart. Any valid RGB color may be used (no CMYK or Pantone colors).
3DBACKSHADE=text	Background shading pattern for the back pane of a standard or 3D chart.
BORDEROPTS=text	Used to supply border and/or background options. You may supply this option more than once. Each one is applied over the last one. By using transparency with inner border options you can provide a glass effect to the chart against the background. Specify the border (top, bottom, left or right) followed by a colon then the size, type, and color. For example: "border:1 solid black" is used for a black border on all sides that is a width of 1 point. Use a semi-colon to separate multiple entries such as "border:1 solid black; border-bottom:2 dashed red;". This example will draw a solid 1 point black border on the top, left and right and a red dashed border that is 2 points thick on the bottom. You may use "border:" to set the options for all sides then use "border-top:", "border-bottom:", "border-left:", and "border-right:" to set any of the other borders as needed. Use a width of 0 to turn off a particular side, such as "border:1 solid black;border-top:0;border-bottom:1 dashed green". The type can be either solid, dotted, dashed, or a pair or numbers to represent the on/off length of a dash. The color can be any valid color setting.

Parameter	Description
	Use the "padding" option to add additional padding to the border. For example, "padding: 5" will add 5 units on all sides. Values for padding do not affect the size or placment of the chart so larger positive values may overlay other areas of the page. Use padding-top, padding-bottom, padding-left, or padding-right to adjust each side separately. You may use the short-hand "padding: top right bottom left;" to set the individual sides as well. For example, "border:1 solid #ccbb99;padding-left:10;padding-right:10;". The padding value can be positive or negative. Positive values will push the border farther away from the center and negative values will move the border closer to the center. Use "background" to set a background color for the border area. For example, "background:#e0e0e0". Use "background-image" to set a background image for the border area. The image will be scaled to fit the border area. For example, "background-image:c:\bkg.jpg". Use "shade" to set a background shading pattern for the border area. For example, "shade:yellowgreen". Use "transparency" to set the transparency name to use for the set. For example, "transparency:mytrans;".
ID=text	An optional ID to assign this chart.

Parameter	Description
HORIZONTAL	Displays basic bar charts horizontally rather than the default vertical format. Only 2D style charts are allowed.
BEZIER	Displays line, area, and ribbon charts using a Bezier curve between the points rather than a straight line. This setting applies to all lines in the chart. You may alternatively set BEZIER on the DATALINE to specify only certain lines should use a Bezier curve.
LEFTDATALABEL=number	Vertically aligns bar values in horizontal bar charts next to the axis line as opposed to having the value labels show to the right of the bar. Specify the distance in points from the axis line.
BGCOLOR=color	Background color for the chart (all charts except PIE type). Any valid RGB color may be used (no CMYK or Pantone colors).
TRANSPARENCY=name	Used along with BGCOLOR to specify a transparency for the base chart background. Use a background color, shading pattern, or image (set with BORDEROPTS) for the main background. This can provide a glass effect to the back of the chart. See the TRANSPARENCY tag for information on setting up a transparency.
BARSTYLE=text	Default 3D bars are rectangular. Values for this option are: Cylinder Pyramid Cone
BARTRANSPARENCY=name	Use to specify a trasnparency for the chart bars or pie slices. See the TRANSPARENCY tag for information on setting up a transparency.

Charts

Parameter	Description
BARDATA=list	A list of numeric values separated by commas for bar charts. Also used as the values for a pie chart. Do not put any special formatting (such as dollar signs or commas).
HIGH=list	A list of numeric values for the high range separated by commas for stock charts. Do not put any special formatting (such as dollar signs or commas).
AVG=list	A list of numeric values for the average range separated by commas for stock charts. Do not put any special formatting (such as dollar signs or commas).
LOW=list	A list of numeric values for the low range separated by commas for stock charts. Do not put any special formatting (such as dollar signs or commas).
STOCKSHADE=color	For STOCK charts only. Draws a horizontal line connecting the HIGH values and one line for the LOW values. The color provided is used to shade the area between the HIGH and LOW values in a STOCK chart. Without this option a vertical line is drawn between each high/low value pair.
DATALAB=list	A list of values separated by a vertical bar (|) to print as the labels above the bars or in pie slices. Use a \n to indicate a new line. For instance, if you have 3 bars you could use Year\n2008|Year\n2009|Year\n2010 to stack "Year" on top of the number for the year.

Parameter	Description
DATALABBGCOLOR=color	Background color for the values in DATALAB. Use this to clear out a rectangle behind the number when drawing grid lines to prevent the grid line from interferring with the number. Any valid RGB color may be used (no CMYK or Pantone colors).
LABFONT=number	Used to specify the font for the data labels.
LABSIZE=number	Used to specify the point size for the data labels.
LABCOLOR=color	Sets text fill color for the data labels.
LABCOMP=number	A percentage to compress the text by. A value less than 100 compresses text while a value greater than 100 expands text.
BARSPACE=number	Used to specify the spacing between bars. The number represents a percentage of the bar size. A value of 10 means take the size of a bar, shrink it by 10% and use that as the spacing.
BARMARGIN=number	Used to specify the amount for the margin between the Y-axis and the first bar. The number represents a percentage of the size of the chart (usually 5 or 10 works good). A value of 5 means out of the size allocated for the chart, leave 5% as the margin.
BARMAX=number	Used to specify the maximum width of a bar. Normally, you'd only use this for a chart with one bar to prevent it's width from taking the entire width of the chart. The number represents a percentage of the size of the chart (usually 5 or 10 works good). A value of 5 means out of the size allocated for the chart, make each bar a maximum of 5%.

Parameter	Description
PIEOFFSET=number	Used to specify an offset amount to break out the pie pieces. The number represents a percentage of the size of the chart (usually 5 or 10 works good).
BARCOLOR=list	A list of comma separated numbers representing a color from 1 to 14 for each bar or pie slice. If not specified, each bar or pie slice is colored starting at 1 for the first, 2 for the second and so on.
BARSHADING=list	A list of comma separated shading names for bar and pie type charts. Use the SHADING tag to setup the shading options.
BARSHADINGTOP	Set this if you want the shading function to run from the top of the chart to the bottom rather than the height of each bar. For example, assume you have a chart that shows monthly sales and you are coloring the top (higher sales) in green and bottom in red. In this case, you probably want to use this option so smaller bars are red while higher ones have more green at the top.
REVEFFECT	Applies shading on the left side of cylinder bar charts rather than the right. This is only when you are not already using a shading pattern.
ALLBARCOLOR=number	Used to set all the bars or lines to the same color (1 to 14). Or, set to an RGB color value such as #3c64a9.
ALLBARSHADING=text	Used to set all the bars or lines to the same shading pattern.
BARPATTERN=list	A list of comma separated numbers representing the patten code from 1 to 8 for each bar or pie slice. See the RECT command and its PATTERN option for a sample of each pattern's appearance.

Parameter	Description
IMAGE=text	Uses the image specified to fill in bars or pie slices. Set IMAGE to the path and file name of the image to use. Images are placed on top of any grid which may be drawn using the YGRID or XGRID option. For this reason, it is best to use images that are as wide as possible so the effect of broken grid lines is minimized. The image is scaled to fit the width of the bar. Use IMAGEFULLBAR or IMAGEFULLCHART to use the full-sized image instead. For pie charts, the image is used full-sized though you can use IMAGESCALE if necessary. Also, the image is always centered at the point of the slice. You may use SLICECOLOR to specify different images for each pie slice.
IMAGEFULLBAR	Use with the IMAGE option. Uses the image without scaling for the bar. The image should be at least as wide as the bar and tall enough for the highest bar. The image starts off each bar in the lower left corner.
IMAGEFULLCHART	Use with the IMAGE option. Uses the image without scaling for the chart. This option allows you to specify an image for the entire chart where each bar shows it's portion of the image. The image should be at least as wide as the chart itself and tall enough for the highest bar. The image starts in the lower left of the chart and each bar shows that section of the image.

Parameter	Description
IMAGESCALE=number	For use with the IMAGEFULLBAR or IMAGEFULLCHART options. The amount to compress or expand the image by in the X and Y direction. Values less than 100 will compress and values greater than 100 will expand. Setting this value overrides any values specified for SCALEX or SCALEY.
IMAGESCALEX=number	For use with the IMAGEFULLBAR or IMAGEFULLCHART options. The amount to compress or expand the image by in the X direction. Values less that 100 will compress and values greater than 100 will expand.
IMAGESCALEY=number	For use with the IMAGEFULLBAR or IMAGEFULLCHART options. The amount to compress or expand the image by in the Y direction. Values less that 100 will compress and values greater than 100 will expand.
IMAGETOP=text	For use with the IMAGE option. This option allows you to specify an image for the top of the bar. Set IMAGETOP to the path and file name of the image to use. This could be rounded cap or other image you want at the top. The width should be the same as the image set with IMAGE but this image should be no more than 20 pixels high as a rough guide.

Parameter	Description
IMAGETOPVALIGN=text	Set to "bottom" to align the bottom of the top image with the bar value. The default without this option is to align the top of the image with the bar value. For example, if the bar value is 50, without this option the top of the image will be at the value 50 on the chart. With this option, the bottom of the image will be at 50 and the top will extend as high as the image.
IMAGEBOT=text	For use with the IMAGE option. This option allows you to specify an image for the bottom of the bar. Set IMAGEBOT to the path and file name of the image to use. This could be rounded cap or other image you want at the bottom. The width should be the same as the image set with IMAGE but this image should be no more than 20 pixels high as a rough guide.
IMAGETOPREV=text	For use with the IMAGE option. This option allows you to specify an image for the top of the bar (highest point) when the bar is drawn below the X axis (negative values). Set IMAGETOPREV to the path and file name of the image to use. This could be rounded cap or other image you want at the bottom - typically a vertically flipped image of IMAGEBOT. The IMAGEBOT is used and flipped vertically if this is not specified and you are using IMAGEBOT. The width should be the same as the image set with IMAGE but this image should be no more than 20 pixels high as a rough guide.

Parameter	Description
IMAGEBOTREV=text	For use with the IMAGE option. This option allows you to specify an image for the bottom of the bar (lowest point) when the bar is drawn below the X axis (negative values). Set IMAGEBOTREV to the path and file name of the image to use. This could be rounded cap or other image you want at the top - typically a vertically flipped image of IMAGETOP. The IMAGETOP is used and flipped vertically if this is not specified and you are using IMAGETOP. The width should be the same as the image set with IMAGE but this image should be no more than 20 pixels high as a rough guide.
LINECOLOR=list	A list of comma separated numbers representing a color from 1 to 14 for each data line. If not specified, each line is colored starting at 1 for the first, 2 for the second and so on.
LINENODE	Use this parameter to turn on drawing of nodes at each point for line charts.
NODEWIDTH=number	The width of stroke line for nodes.
LINELAB	Sets the XLABEL and DATAMLAB to align with line points rather than bars. Use this option when you're showing a line chart and want to show the data values along the bottom.
COMPACTLEGEND	Draws a compact legend without data values for line charts. Use the LEGENDCOLS option to specify the number of columns you want to have. The default is to have as many columns as there are data lines. Sepcify the LEGEND in the DATALINE tag for each line.

Parameter	Description
LEGEND=list	Specifies the labels to use in the legend. Separate each value by a vertical bar (|) and use \n as the new-line character.
LEGENDBGCOLOR=color	Background color for the chart legend. Any valid RGB color may be used (no CMYK or Pantone colors).
LEGENDSHADE=text	Background shading pattern for the chart legend.
LEGENDOFFSET=number	Used to position the legend if the default position is not where you want it. Use a positive or negative number (usually in the range of -5 to 5) to position the legend.
LEGENDLABWIDTH=number	The width to use for the legend label when using DATAMLAB. The default is 15.
LEGENDCOLS=number	Sets the number of columns to divide the legend into when using with pie charts. Default value is 1. Also used with the COMPACTLEGEND option for line charts.
LEGENDPCT=number	Sets aside the percent of the total width for legends when using pie charts. For example, set to 40 to use 40% of the width for the legend and the other 60% to the left will be used for the chart. Using this option will cause text to wrap when necessary in the legend. This could result in text flowing below the area set aside for the chart. You'll need to add in your own line breaks in the legend, if necessary, if you do not use this option with pie charts.
LGLINESPACE=number	A number to increase or decrease the legend line spacing by when using line charts along with DATAMLAB or the LEGEND option on DATALINE.

Charts

Parameter	Description
NODE=number	The symbol to draw for scatter charts or the average point for stock charts. A solid square is used for scatter and a horizontal line for stock by default. Use a number from 1 to 12.
NODESIZE=number	Used to indicate the size of the nodes drawn for scatter and stock charts. NODESIZE should be set to a number between 1 and 10.
NODEFCOLOR=color	Sets the node fill color for scatter and stock charts (default is FCOLOR).
NODESCOLOR=color	Sets the node stroke color for scatter and stock charts (default is SCOLOR). Only for nodes that are filled or use for the AVG line color in STOCK charts when using the STOCKSHADE option.
BASEFCOLOR=color	Sets the base fill color for axis lines and other base elements. The default is black (#000000).
BASESCOLOR=color	Sets the base stroke color for axis lines and other base elements. The default is black (#000000).
XGRID=number	Used mainly for line charts. Specifies the number of equally spaced vertical lines to draw.
YGRID=number	Specifies the number of equally spaced horizontal lines to draw. Will be set automatically if YSCALE is not used. Set YGRID=0 If YSCALE is not used and you don't want lines drawn.
GRIDON=number	The length of the line segment to draw with the SCOLOR for the grid.
GRIDOFF=number	The length of the line segment to draw with the FCOLOR for the grid.

Parameter	Description
GRIDDASH=number,number [,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length for the grid.
GRIDFCOLOR=color	Sets the grid fill color.
GRIDSCOLOR=color	Sets the grid stroke color.
SHOWXTICKS=On|Off	Sets the display of the X axis tick marks.
XSCALE=list	Used for scatter charts to specifiy the minimum and maximum values for the X-axis. Enter a pair of values separated by a comma. To use from -50 to 100 enter -50,100.
YSCALE=list	Used for all charts except pie to specifiy the minimum and maximum values for the Y-axis. Enter a pair of values separated by a comma. To use from -50 to 100 enter -50,100. You may leave this option off and the program will compute a set of values to use.
YSCALE2=list	Used for line charts to set the scale for a second Y-axis. Enter a pair of values separated by a comma. To use from -50 to 100 enter -50,100.
YSCALEPFX=text	Text to place in front of the values when YSCALE is left off and the program computes the value. For example, use $ to place a dollar sign in front of the numbers on the Y-axis. This is only used when YSCALE is computed by the program.
YSCALESFX=text	Text to place after the values when YSCALE is left off and the program computes the value. For example, use K to place a the thousands symbol after the numbers on the Y-axis. This is only used when YSCALE is computed by the program.

Parameter	Description
DATAMULT=number	Used by bar charts to specify the number of values being grouped. For example, you may be listing actual vs. project on the same chart broken out by month. In this case you would use 2 as this value. Then for BARDATA you would enter actual-month1, projected-month1, actual-month2, projected-month2, etc. Alternatively, you may use the BARDATAMULT tag to enter groups of data. You do not need to specify this option in that case as the number of BARDATAMULT tags will determine this value.
DATAMLAB=list	Used to specify the label for each group when using DATAMULT. Separate the values by a vertical bar (|). If DATAMULT is set to 2 you should have two text items here with a | separating them.
VALUETOP	Places the data values (specified in DATALAB) above each bar when using DATAMULT or BARDATAMULT. Normally, the values are placed below the bar in a separate list when using these options.
BARCOLUMNS	Used with the DATAMULT option or BARDATAMULT tags and the 3D option. Places the columns in back of one another rather than next to each other.

Parameter	Description
LEGENDCOLOR=list	A list of comma separated numbers representing a color from 1 to 14 for the legend when using the DATAMULT option. If not specified, the colors used in the legend are taken from the beginning of the list in BARCOLOR. The only time this should be used is when the first n entries in BARCOLOR don't correspond the colors you want used in the legend (perhaps the first few bars have a special meaning from the rest of the bars and are colored differently).
NOCOLORLEGEND	Do not show the color blocks in the legend for bar charts using the DATAMULT or BARDATAMULT option.
MULTCOLORLEGEND	Show the color legend next to the values for bar charts using the DATAMULT or BARDATAMULT option.
OUTSIDELABELS	Used to force all labels for pie charts outside of the pie slice.
XSCALEFONT=number	Used to specify the font for the X-axis scale labels.
XSCALESIZE=number	Used to specify the point size for the X-axis scale labels.
XSCALECOLOR=color	Sets text fill color for the X-axis scale labels.
XSCALECOMP=number	A percentage to compress the X-axis scale labels text by. A value less than 100 compresses text while a value greater than 100 expands text.
YSCALEFONT=number	Used to specify the font for the Y-axis scale labels.
YSCALESIZE=number	Used to specify the point size for the Y-axis scale labels.
YSCALECOLOR=color	Sets text fill color for the Y-axis scale labels.

Parameter	Description
YSCALECOMP=number	A percentage to compress the Y-axis label text by. A value less than 100 compresses text while a value greater than 100 expands text.
XAXISLABEL=text	Used to specify the label for the X-axis.
XAXISFONT=number	Used to specify the font for the X-axis label.
XAXISSIZE=number	Used to specify the point size for the X-axis label.
XAXISCOLOR=color	Sets text fill color for the X-axis labels.
XAXISCOMP=number	A percentage to compress the X-axis label text by. A value less than 100 compresses text while a value greater than 100 expands text.
XAXISOFFSET=number	Used to position the X-axis label if the default position is not where you want it. Use a positive or negative number (usually in the range of -5 to 5) to position the label.
YAXISLABEL=text	Used to specify the label for the Y-axis.
YAXISFONT=number	Used to specify the font for the Y-axis label.
YAXISSIZE=number	Used to specify the point size for the Y-axis label.
YAXISCOLOR=color	Sets text fill color for the Y-axis labels.
YAXISCOMP=number	A percentage to compress the Y-axis label text by. A value less than 100 compresses text while a value greater than 100 expands text.
YAXISOFFSET=number	Used to position the Y-axis label if the default position is not where you want it. Use a positive or negative number (usually in the range of -5 to 5) to position the label.

Parameter	Description
YAXISOFFSET2=number	Used to position the 2nd Y-axis label if the default position is not where you want it. Use a positive or negative number (usually in the range of -5 to 5) to position the label.
XLABEL=list	Used to specify the labels for the X-axis in charts. Separate each entry with a vertical bar (|). Use \n for a new-line.
XLABELFONT=number	Used to specify the font for the X-axis labels.
XLABELSIZE=number	Used to specify the point size for the X-axis labels.
XLABELCOLOR=color	Sets text fill color for the X-axis labels.
XLABELCOMP=number	A percentage to compress the X-axis labels text by. A value less than 100 compresses text while a value greater than 100 expands text.
YLABEL=list	Used to specify the labels for the Y-axis in charts. Separate each entry with a vertical bar (|). Use \n for a new-line. You should leave this off if YSCALE is not used. The program will assign values automatically in this case.
YLABEL2=list	Used to specify the labels for the 2nd Y-axis in charts. These labels print on the right hand side of the chart. Separate each entry with a vertical bar (|). Use \n for a new-line.
XLABELOFFSET=number	Used to position the X-axis labels up or down if the default position is not where you want it. Use a positive or negative number (usually in the range of -5 to 5) to position the label. Positive numbers move the lables down and negative move them up.

Parameter	Description
YLABELOFFSET=number	Used to position the Y-axis labels right or left if the default position is not where you want it. Use a positive or negative number (usually in the range of -5 to 5) to position the label. Positive numbers move the lables right and negative move them left.
XLABELOFFSET2=number	Used to position the X-axis labels right or left if the default position is not where you want it. Use a positive or negative number (usually in the range of -5 to 5) to position the label. Positive numbers move the lables right and negative move them left.
YLABELOFFSET2=number	Used to position the Y-axis labels up or down if the default position is not where you want it. Use a positive or negative number (usually in the range of -5 to 5) to position the label. Positive numbers move the lables down and negative move them up.
XLABELPOS=number	The Y-axis value to display the XLABEL list at. Normally the XLABEL list is displayed at the bottom of the chart. For example, if your YSCALE=-50,50 then setting XLABELPOS=0 will display the X-axis labels at the Y=0 position - half of the height of the chart. This option is not for bar charts.
YLABELPOS=number	The X-axis value to display the YLABEL list at. Normally the YLABEL list is displayed at the right of the chart. For example, if your XSCALE=-50,50 then setting YLABELPOS=0 will display the Y-axis labels at the X=0 position - half of the width of the chart. This option is not for bar charts.

Parameter	Description
TOPLABELS	Used to specify that XLABEL and YLABEL text be drawn on top of other chart items. Only useful when using the XLABELPOS or YLABELPOS settings and you want the axis labels drawn on top of any chart graphics.
LABANGLE=number	Used to set the angle in degrees for the labels on the X-axis. Typically this would be set to a value between 30 and 90.
FONT=number	Used to specify the font for the axis labels.
SIZE=number	Used to specify the point size for the axis labels.
COLOR=color	Sets text fill color for the axis labels.
COMP=number	A percentage to compress the axis label text by. A value less than 100 compresses text while a value greater than 100 expands text.
TITLE=text	Used to specify a chart title. Use \n for a new-line.
TITLEFONT=number	Used to specify the font for the chart title.
TITLESIZE=number	Used to specify the point size for the chart title.
TITLECOLOR=color	Sets text fill color for the chart title.
TITLECOMP=number	A percentage to compress the chart title text by. A value less than 100 compresses text while a value greater than 100 expands text.
TITLEOFFSET=number	Used to position the chart title if the default position is not where you want it. Use a positive or negative number (usually in the range of -2 to 2) to position the title up or down.

<SLICECOLOR
VALUE=color
BORDER=color
IMAGE=text
SCALE=number
SCALEX=number
SCALEY=number>

Used to define a color for a pie chart or slice background image. Repeat the tag for as many colors as you need setting each one to the next color to use.

Parameter	Description
VALUE=color	Any valid RGB color code.
BORDER=color	Any valid RGB color code. The border is used with the SLICEBORDER option and will not draw a border by simply setting this value.
IMAGE=text	Uses the image specified to fill in the pie slice. Set IMAGE to the path and file name of the image to use. For pie charts, the image is used full-sized though you can use IMAGESCALE if necessary. Also, the image is always centered at the point of the slice. Be sure to use an image big enough or scale it to cover the entire pie chart so each slice can properly display its portion of the image.
SCALE=number	Optional. The amount to compress or expand the image by in the X and Y direction. Values less than 100 will compress and values greater than 100 will expand. Setting this value overrides any values specified for SCALEX or SCALEY.
SCALEX=number	Optional. The amount to compress or expand the image by in the X direction. Values less that 100 will compress and values greater than 100 will expand.
SCALEY=number	Optional. The amount to compress or expand the image by in the Y direction. Values less that 100 will compress and values greater than 100 will expand.

<BARBORDER
STRENGTH="number"
WIDTH="number"
COLOR="color">

Used to define border options for each bar on bar charts. The color of the border is based on the color of the bar. To force a particular color for each bar, set the COLOR option.

Parameter	Description
STRENGTH	The strength of the border based on the bar color. Set to a value between -100 and 100. Negative values will darken the border effect and positive values will lighten. The color used is based on the color of the bar. Use the COLOR option to force a particular color.
WIDTH	The width of the border around the bar to draw in points. The default is 0 or no border.
COLOR	The color for the border around the bar. Any valid RGB color may be used (no CMYK or Pantone colors). Or, use STRENGTH to set to a color lighter or darker based on the bar color.

<SLICEBORDER
ARC
SIDES="text"
STRENGTH="number"
WIDTH="number"
SHADE>

Used to define border options for each slice on pie charts. The color of the border or shading is based on the color of the slice the effect is applied to. To force a particular color for a slice, set the BORDER option on the SLICECOLOR tag.

Parameter	Description
ARC	Set this to apply the border options to the outer arc of the slice.
SIDES="text"	Set to "1" to apply the border options to one edge of the slice. Set to "2" to apply to the opposite edge. Set to "1,2" to apply to both edges. When using with ARC, both "1,2" will apply.
STRENGTH=number	The strength of the effect. Set to a value between -100 and 100. Negative values will darken the border effect and positive values will lighten. The color used is based on the color of the slice or the BORDER value from the SLICECOLOR tag.
WIDTH=number	The width of the border effect. This is units of points (1/72 of an inch).
SHADE	Transition the effect as a gradient rather than a solid color. This can be used to provide a shadow effect at the edges of pie slices.

<DATALINE
LINE=number
VALUES=list
LABELS=list
LEGEND=text
YAXIS=number
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
BEZIER
NODE=file|number
NODESIZE=number
NODEFCOLOR=color
NODESCOLOR=color
FCOLOR=color
SCOLOR=color>

Used to define the values for each line in a line (BAR-LINE) or each set of points in a scatter chart. Each line or point group to be plotted will have a separate DATALINE tag. This tag goes between the opening and closing CHART tag. To leave out a data point (or points) for a given line don't put anything between the commas where that value would go. For example, if you have three lines but don't want to plot all points for all lines do something like this:

```
<DATALINE LINE=1 VALUES=15,23,45,42,21>
<DATALINE LINE=2 VALUES=,,46,58,18>
<DATALINE LINE=3 VALUES=12,43,,,>
```

The first line will show all five values, line 2 will only plot three points and the last line will have two points.

Parameter	Description
LINE=number	The line number. Start at 1 for the first line. Do not reuse a number in the same chart or the data will be overwritten.
VALUES=list	The list of values separated by a comma. Do not put in any special formatting like dollar signs or commas.
LABELS=list	The list of lables separated by a vertical bar (|).

Parameter	Description
LEGEND=text	The text to show in the legend. If you place the label here you do not need to use DATAMLAB.
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 on any line set where the values reflect the scaling factor from the secondary Y-axis.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
BEZIER	Use a Bezier curve for the line segments between the points rather than a straight line.
NODE=file|number	The symbol to draw at the data points (box, triangle, etc). The symbol to use is determined by the software if not specified. Use a number from 1 to 12. Or, specify an image file name. The image will be used as the node in this case.
NODESIZE=number	Used to indicate the size of the nodes drawn. NODESIZE should be set to a number between 1 and 10.
NODEFCOLOR=color	Sets the node fill color (default is FCOLOR).
NODESCOLOR=color	Sets the node stroke color (default is SCOLOR). Only for nodes that are filled.

Parameter	Description
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.

<BARSERIES
VALUES=list
FCOLOR=color
FCOLORLIST=color[,color...]
PATTERN=number>

Used to define the values for a stacked bar chart. Each item with a series of values will have a separate BARSERIES tag. This tag goes between the opening and closing CHART tag.

Parameter	Description
VALUES=list	The list of values separated by a comma. Do not put in any special formatting like dollar signs or commas.
FCOLOR=color	Sets the fill color.
FCOLORLIST=color[,color]	Used to set a different fill color for each bar. Set to a comma separated list of color names or specify in hex format.
PATTERN=number	A number representing the patten code from 1 to 8 for the series. See the RECT command and its PATTERN option for a sample of each pattern's appearance.

<BARDATAMULT
VALUES=list
LABELS=list
LEGEND=text
FCOLOR=color
PATTERN=number
SHADING=text>

Used to set up a group of values for a bar chart with multiple groups. Each group of values will have a separate BARSERIES tag. This tag goes between the opening and closing CHART tag.

Parameter	Description
VALUES=list	The list of values separated by a comma. Do not put in any special formatting like dollar signs or commas.
LABELS=list	The list of label values separated by a vertical bar (|).
LEGEND=text	The text to show in the legend.
FCOLOR=color	Sets the fill color.
PATTERN=number	A number representing the patten code from 1 to 8 for the bar group. See the RECT command and its PATTERN option for a sample of each pattern's appearance.
SHADING=text	The shading pattern to use for the bar group.

<CHARTCOLOR
COLORNUM=number
COLOR=color>

Used to define a custom chart color. There are 14 default colors used by the charts. You can use this command to redefine any of the 14 colors. For example, the first bar on a bar chart uses color 1 (unless otherwised specified with the BARCOLOR option on the PARAMS tag). The next bar uses color 2, and so on. This command can be used to redefine what colors 1 through 14 are. Any valid RGB color may be used (no CMYK or Pantone colors). The color settings will remain in effect until set to another value or cleared out (by leaving off the COLOR option from this tag).

Parameter	Description
COLORNUM=number	The color number to change. This can be a value from 1 to 14.
COLOR=color	The color to use for this COLORNUM.

<CHARTPOINTS
VAR=[X|Y]
POINT=[M|A]
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
FCOLOR=color
SCOLOR=color
NODESIZE=number
NODE=number
NODEFCOLOR=color
NODESCOLOR=color>
x1, y1 [, x2, y2, ... xn, yn]
</CHARTPOINTS>

Used to draw points on the chart and optionally create a line histogram. Must use both the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the YSCALE goes from 0 to 50 then the Y values should be between 0 and 50.

Place any number of x, y pairs between the opening and closing tag. You may separate the values with commas or spaces or any combination of the two. In addition, you may include line breaks within the data section.

Use the VAR and POINT options to generate a line based histogram. This type of chart plots points and combines like values for one of the axes. A line is drawn between the endpoints for each of the like values and a line across the entire chart though the mid-points. See an example here.

Parameter	Description
VAR=[X|Y]	Optional. Used to specify the axis with the variable values. Use this parameter when one of the axis values contains several values for the same point on the other axis. For example, if you have several X values that all pertain to a given Y value, then set VAR=X.

Parameter	Description
POINT=[M|A]	Optional. Set to M for mid-point or A for average. Used to specify the function to compute the X or Y value to draw a line through for histograms. For example, if you have X values of 1, 1.5, and 1.75 then setting POINT=M will use the X value 1.375 (which is 1 + (1.75 - 1) / 2) for the main line. Using POINT=A for the same values will cause the X value to compute as 1.4166 (which is (1 + 1.5 + 1.75) / 3).
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
NODESIZE=number	Used to indicate the size of the nodes drawn for the points. NODESIZE should be set to a number between 1 and 10.
NODEFCOLOR=color	Sets the node fill color for the points (default is FCOLOR).
NODESCOLOR=color	Sets the node stroke color for the points (default is SCOLOR). Only for nodes that are filled.

<CHARTLINE
X1=number
Y1=number
X2=number
Y2=number
YAXIS=number
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
FCOLOR=color
SCOLOR=color
BACKGROUND
ZORDER=number>

Used to draw a line on the chart. Must use both the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the YSCALE goes from 0 to 50 then the Y values should be between 0 and 50.

Parameter	Description
X1=number	From X point based on the chart's XSCALE.
Y1=number	From Y point based on the chart's YSCALE.
X2=number	Thru X point based on the chart's XSCALE.
Y2=number	Thru Y point based on the chart's YSCALE.
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 when the values reflect the scaling factor from the secondary Y-axis.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.

Parameter	Description
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
BACKGROUND	Draws the line first, before the chart is drawn if this is set.
ZORDER=number	The order to draw the line in releation to the other CHART drawing commands. Lower numbered objects are drawn first.

```
<CHARTSERIES
    OVER=number
    UNDER=number
    YAXIS=number
    ON=number
    OFF=number
    DASH=number,number[,number,number...]
    WIDTH=number
    FCOLOR=color
    SCOLOR=color
    BACKGROUND
    ZORDER=number>
</CHARTSERIES>
```

Used to draw a series of lines on the chart and optionally highlight over/under regions. Must use both the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the YSCALE goes from 0 to 50 then the Y values should be between 0 and 50. Place the data points between the CHARTSERIES and /CHARTSERIES tags. You may split them up on as many lines as necessary. Separate values on the same line with a comma. Data should be entered in the form X1,Y1,X2,Y2,etc.

Parameter	Description
OVER=number	Optional. The Y point to use as a cutoff for the maximum value. Any areas of the series that go over this point will be filled in with the color specified by FCOLOR.
UNDER=number	Optional. The Y point to use as a cutoff for the minimum value. Any areas of the series that go under this point will be filled in with the color specified by FCOLOR.
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 when the values reflect the scaling factor from the secondary Y-axis.
ON=number	The length of the line segment to draw with the SCOLOR.

Parameter	Description
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
BACKGROUND	Draws the series first, before the chart is drawn if this is set.
ZORDER=number	The order to draw the series in releation to the other CHART drawing commands. Lower numbered objects are drawn first.

<CHARTRECT
X1=number
Y1=number
X2=number
Y2=number
YAXIS=number
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
FILL
FCOLOR=color
SCOLOR=color
BACKGROUND
ZORDER=number>

Used to draw a rectangle on the chart. Must use both the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the YSCALE goes from 0 to 50 then the Y values should be between 0 and 50.

Parameter	Description
X1=number	From X point based on the chart's XSCALE.
Y1=number	From Y point based on the chart's YSCALE.
X2=number	Thru X point based on the chart's XSCALE.
Y2=number	Thru Y point based on the chart's YSCALE.
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 when the values reflect the scaling factor from the secondary Y-axis.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.

Parameter	Description
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FILL	If set fills the rectangle with the fill color.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
BACKGROUND	Draws the rectangle first, before the chart is drawn if this is set.
ZORDER=number	The order to draw the rectangle in releation to the other CHART drawing commands. Lower numbered objects are drawn first.

<CHARTPOLY
POINTS=number,number[,number]
YAXIS=number
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
FILL
FCOLOR=color
SCOLOR=color
BACKGROUND
ZORDER=number>

Used to draw a polygon on the chart. Must use both the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the YSCALE goes from 0 to 50 then the Y values should be between 0 and 50.

Parameter	Description
POINTS=number, number [,number]	Series of points in the form X1,Y1,X2,Y2,... as based on the current chart XSCALE and YSCALE settings.
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 when the values reflect the scaling factor from the secondary Y-axis.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FILL	If set fills the rectangle with the fill color.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.

Parameter	Description
BACKGROUND	Draws the polygon first, before the chart is drawn if this is set.
ZORDER=number	The order to draw the polygon in releation to the other CHART drawing commands. Lower numbered objects are drawn first.

<CHARTCIRCLE
X=number
Y=number
RADIUS=number|numberI
YAXIS=number
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
FILL
FCOLOR=color
SCOLOR=color
BACKGROUND
ZORDER=number>

Used to draw a circle on the chart. Must use both the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the YSCALE goes from 0 to 50 then the Y values should be between 0 and 50.

Parameter	Description
X=number	X point based on the chart's XSCALE.
Y=number	Y point based on the chart's YSCALE.
RADIUS=number|numberI	Radius based on the chart's XSCALE. If you put a letter I after the number then radius is expressed in inches.
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 when the values reflect the scaling factor from the secondary Y-axis.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.

Parameter	Description
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FILL	If set fills the circle with the fill color.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
BACKGROUND	Draws the circle first, before the chart is drawn if this is set.
ZORDER=number	The order to draw the circle in releation to the other CHART drawing commands. Lower numbered objects are drawn first.

<CHARTCURVE
X1=number
Y1=number
X2=number
Y2=number
CPX1=number
CPY1=number
CPX2=number
CPY2=number
YAXIS=number
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
FCOLOR=color
SCOLOR=color
BACKGROUND
ZORDER=number>

Used to draw a curve on the chart. Must use both the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the YSCALE goes from 0 to 50 then the Y values should be between 0 and 50.

Parameter	Description
X1=number	First X point based on the chart's XSCALE.
Y1=number	First Y point based on the chart's YSCALE.
X2=number	Second X point based on the chart's XSCALE.
Y2=number	Second Y point based on the chart's YSCALE.
CPX1=number	X position of first control point.
CPY1=number	Y position of first control point.
CPX2=number	Optional. X position of second control point.
CPY2=number	Optional. Y position of second control point.

Parameter	Description
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 when the values reflect the scaling factor from the secondary Y-axis.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
BACKGROUND	Draws the curve first, before the chart is drawn if this is set.
ZORDER=number	The order to draw the curve in releation to the other CHART drawing commands. Lower numbered objects are drawn first.

<CHARTNORMDIST
X1=number
X2=number
MU=number
SIGMA=number
SCALE=number
WIDTH=number
SCOLOR=color
YAXIS=number
BACKGROUND
ZORDER=number>

Used to draw a normal distribution (bell curve) on the chart. Must use the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the XSCALE goes from 0 to 50 then the X values should be between 0 and 50. The formula for a normal distribution is $\frac{1}{\sqrt{2\pi\sigma}} e^{-(x-\mu)^2/2\sigma^2}$ where μ is equal to the mean and σ is equal to the standard deviation.

Parameter	Description
X1=number	First X point based on the chart's XSCALE.
X2=number	Last X point based on the chart's XSCALE.
MU=number	The mean value.
SIGMA=number	The standard deviation value.
SCALE=number	A scaling factor to apply to the normal distribution. All Y values are multiplied by this value before plotting. Default is 1.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
SCOLOR=color	Sets the stroke color.
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 when the values reflect the scaling factor from the secondary Y-axis.
BACKGROUND	Draws the normal distribution first, before the chart is drawn if this is set.
ZORDER=number	The order to draw the normal distribution in releation to the other CHART drawing commands. Lower numbered objects are drawn first.

<CHARTTEXT
X=number
Y=number
ALIGN=L|R|C
FACE=font number
SIZE=point size
FCOLOR=color
SCOLOR=color
YAXIS=number
BACKGROUND
ZORDER=number>
</CHARTTEXT>

Used to place text on the chart. The actual text to print goes between the opening CHARTTEXT and closing /CHARTTEXT tags. Must use both the YSCALE and XSCALE parameters when setting up the chart. The X/Y coordinates are in terms of these values. For example, if the YSCALE goes from 0 to 50 then the Y values should be between 0 and 50. You may use BR tags inside the text itself but there is no autowrapping of text.

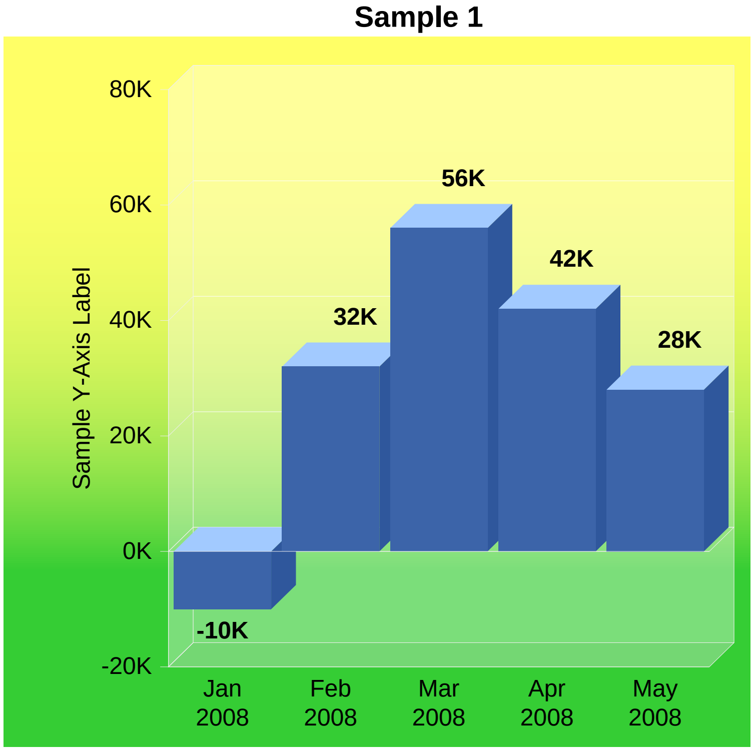
Parameter	Description
X=number	X point based on the chart's XSCALE.
Y=number	Y point based on the chart's YSCALE. The Y value is used for the baseline of the text.
ALIGN=L|R|C	Sets the initial alignment to Left, Right or Center. Default is Left.
FACE=font number	Sets the font.
SIZE=point size	Sets the point size for the font.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
YAXIS=number	Default is primary Y-axis. You may set a secondary Y-axis by using the YSCALE2 option in a chart PARAMS tag. Set YAXIS=2 when the values reflect the scaling factor from the secondary Y-axis.
BACKGROUND	Draws the text first, before the chart is drawn if this is set.
ZORDER=number	The order to draw the text in releation to the other CHART drawing commands. Lower numbered objects are drawn first.

<RADARSERIES
VALUES=list
LEGEND=text
FCOLOR=color
SCOLOR=color
FILL
WIDTH=number>

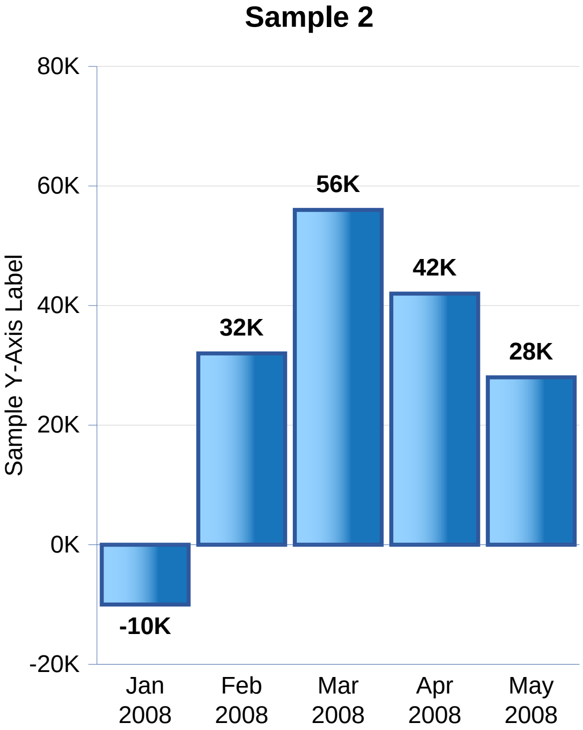
Used to set up a group of values for a radar chart. Each group of values will have a separate RADARSERIES tag. This tag goes between the opening and closing CHART tag.

Parameter	Description
VALUES=list	The list of values separated by a comma. Do not put in any special formatting like dollar signs or commas. Specify one value for each XLABEL entry. For example, if you have all 12 months as your labels, you should have 12 values.
LEGEND=text	The text to show in the legend.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the line color.
FILL	If set fills the area with the fill color.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.

Charts

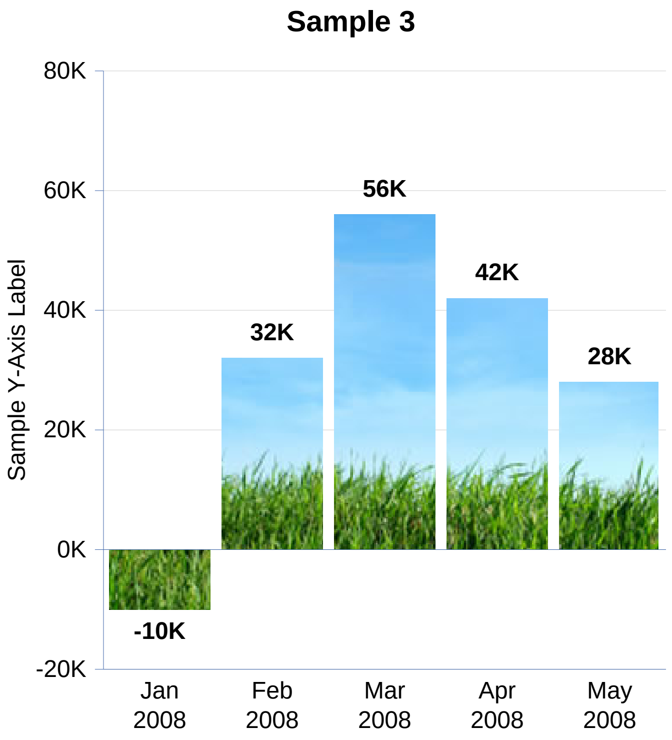


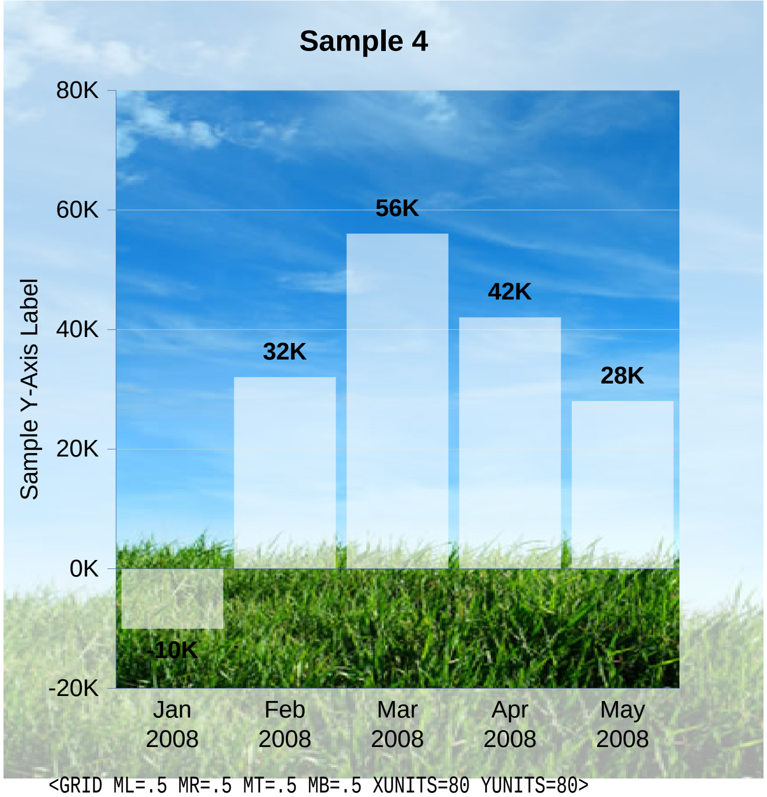
```
<SHADING NAME="yellgreen" COLOR1=#FFFF66 COLOR2=#33CC33
COLORARY="0,.25,0,1">
<TRANSPARENCY NAME="trans1" VALUE=65>
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART HEIGHT=35 WIDTH=35>
<PARAMS TYPE="BAR-LINE">
<PARAMS TITLE="Sample 1" TITLEFONT=5 TITLESIZE=11>
<PARAMS 3D=15 3DTOP=40 3DSIDE=-5>
<PARAMS BARDATA=-10,32,56,42,28>
<PARAMS DATALAB="-10K|32K|56K|42K|28K">
<PARAMS YSCALE=-20,80>
<PARAMS YLABEL="-20K|0K|20K|40K|60K|80K">
<PARAMS BARSPACE=10 BASESCOLOR=#eee GRIDSCOLOR=#fff>
<PARAMS YGRID=5 BGCOLOR=#FFF>
<PARAMS BORDEROPTS="padding:5 3 18 35;shade:yellgreen">
<PARAMS ALLBARCOLOR=#3c64a9 LABFONT=5 XLABELFONT=2>
<PARAMS YAXISLABEL="Sample Y-Axis Label">
<PARAMS YAXISOFFSET=5 TRANSPARENCY=trans1>
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|May\n20
08">
<PARAMS FONT=2 SIZE=9>
</CHART>
```



```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<SHADING NAME="wb" COLOR1=#95d1ff COLOR2=#1774ba
COLORARY="0,0,.65,0">
<CHART HEIGHT=35 WIDTH=30>
<PARAMS TYPE="BAR-LINE">
<PARAMS TITLE="Sample 2" TITLEFONT=5 TITLESIZE=11>
<BARBORDER WIDTH=1.5 STRENGTH=-5>
<PARAMS BARDATA=-10,32,56,42,28>
<PARAMS DATALAB="-10K|32K|56K|42K|28K">
<PARAMS YSCALE=-20,80>
<PARAMS YLABEL="-20K|0K|20K|40K|60K|80K">
<PARAMS BARSPACE=10 BASESCOLOR=#3c64a9
GRIDSCOLOR=#ccc>
<PARAMS YGRID=5 ALLBARSHADING=wb>
<PARAMS ALLBARCOLOR=#3c64a9 LABFONT=5
XLABELFONT=2>
<PARAMS YAXISLABEL="Sample Y-Axis Label">
<PARAMS YAXISOFFSET=6>
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=2 SIZE=9>
```

Charts

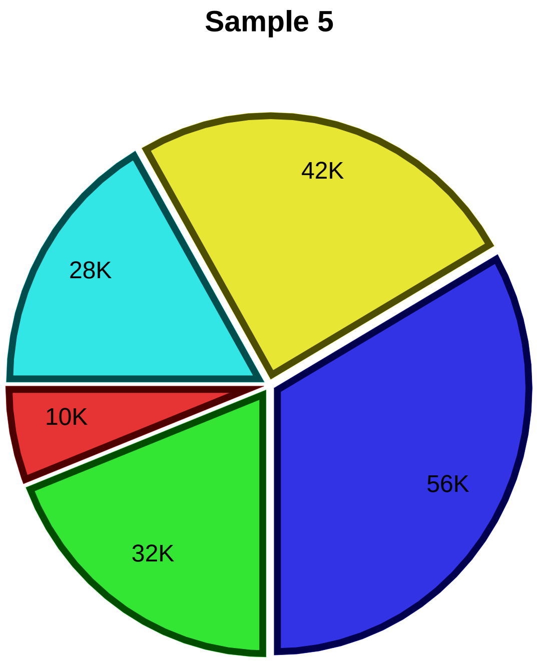


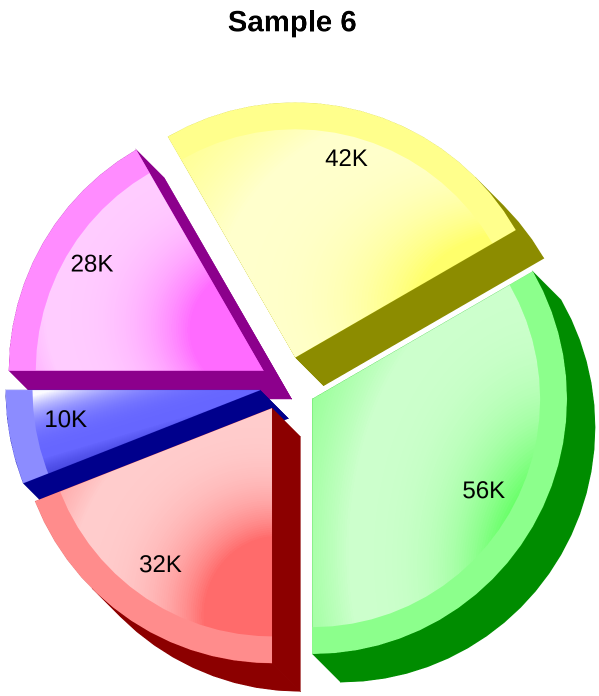


```
<SHADING NAME="yellgreen" COLOR1=#FFFF66 COLOR2=#33CC33
COLORARY="0,.25,0,1">
<TRANSPARENCY NAME="trans1" VALUE=65>
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART HEIGHT=35 WIDTH=35>
<PARAMS TYPE="BAR-LINE">
<PARAMS TITLE="Sample 3" TITLEFONT=5 TITLESIZE=11>
<PARAMS IMAGE="blue_sky_grass.jpg" IMAGEFULLCHART
IMAGESCALE=65>
<PARAMS BARDATA=-10,32,56,42,28>
<PARAMS DATALAB="-10K|32K|56K|42K|28K">
<PARAMS YSCALE=-20,80>
<PARAMS YLABEL="-20K|0K|20K|40K|60K|80K">
<PARAMS BARSPACE=10 BASESCOLOR=#3c64a9 GRIDSCOLOR=#ccc>
<PARAMS YGRID=5>
<PARAMS ALLBARCOLOR=#3c64a9 LABFONT=5 XLABELFONT=2>
<PARAMS YAXISLABEL="Sample Y-Axis Label">
<PARAMS YAXISOFFSET=5>
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|May\n20
08">
<PARAMS FONT=2 SIZE=9>
</CHART>
```

```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<SHADING NAME="wb" COLOR1=#95d1ff COLOR2=#1774ba
COLORARY="0,0,.65,0">
<CHART HEIGHT=35 WIDTH=35>
<PARAMS BORDEROPTS="padding:15;padding-
left:20;background-
image:blue_sky_grass.jpg;transparency:trans1"">
<PARAMS BORDEROPTS="background-
image:blue_sky_grass.jpg">
<PARAMS BARDATA=-10,32,56,42,28>
<PARAMS DATALAB="-10K|32K|56K|42K|28K">
<PARAMS YSCALE=-20,80>
<PARAMS YLABEL="-20K|0K|20K|40K|60K|80K">
<PARAMS BARSPACE=10 BASESCOLOR=#3c64a9
GRIDSCOLOR=#ccc>
<PARAMS YGRID=5>
<PARAMS ALLBARCOLOR=#fff LABFONT=5 XLABELFONT=2
BARTRANSPARENCY=trans2>
<PARAMS YAXISLABEL="Sample Y-Axis Label">
<PARAMS YAXISOFFSET=5>
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=2 SIZE=9>
</CHART>
```

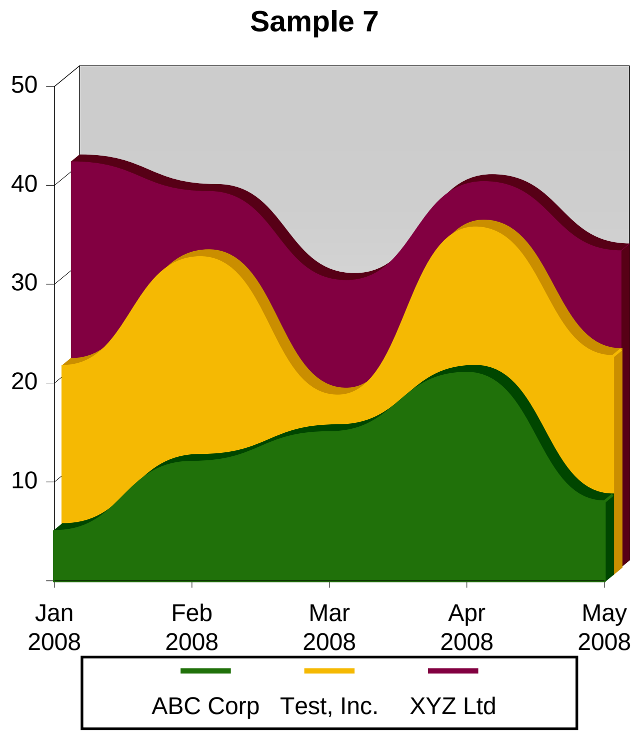
Charts



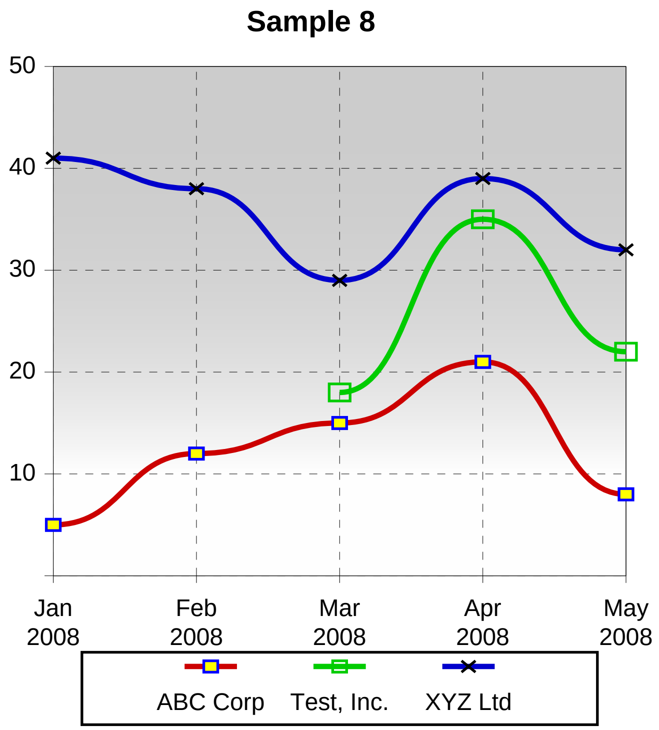


```
<SHADING NAME="yellgreen" COLOR1=#FFFF66
COLOR2=#33CC33 COLORARY="0,.25,0,1">
<TRANSPARENCY NAME="trans1" VALUE=65>
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART HEIGHT=35 WIDTH=35>
<PARAMS TYPE="PIE">
<PARAMS TITLE="Sample 5" TITLEFONT=5 TITLESIZE=11>
<SLICEBORDER ARC SIDES="1,2" STRENGTH=-60 WIDTH=5>
<PARAMS BARDATA=10,32,56,42,28>
<PARAMS DATALAB="10K|32K|56K|42K|28K">
<PARAMS FONT=2 SIZE=9 PIEOFFSET=1>
</CHART>
```

```
<SHADING NAME="b" COLOR1=#3333CC COLOR2=#6666FF
COLOR3=#9999FF COLOR4=#6666FF
COLORARY=".5,.5,2,1,.2,.3">
<SHADING NAME="r" COLOR1=#CC3333 COLOR2=#FF6666
COLOR3=#FF9999 COLOR4=#FF6666
COLORARY=".5,.5,2,1,.2,.3">
<SHADING NAME="g" COLOR1=#33CC33 COLOR2=#66FF66
COLOR3=#99FF99 COLOR4=#66FF66
COLORARY=".5,.5,2,1,.2,.3">
<SHADING NAME="y" COLOR1=#CCCC33 COLOR2=#FFFF66
COLOR3=#FFFF99 COLOR4=#FFFF66
COLORARY=".5,.5,2,1,.2,.3">
<SHADING NAME="p" COLOR1=#CC33CC COLOR2=#FF66FF
COLOR3=#FF99FF COLOR4=#FF66FF
COLORARY=".5,.5,2,1,.2,.3">
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=40 X2=75 HEIGHT=35>
<PARAMS TYPE="PIE">
<PARAMS TITLE="Sample 6" TITLEFONT=5 TITLESIZE=11>
<SLICEBORDER ARC STRENGTH=35 WIDTH=20>
<PARAMS BARDATA=10,32,56,42,28>
<PARAMS DATALAB="10K|32K|56K|42K|28K">
<PARAMS BARSHADING=b,r,g,y,p>
<PARAMS PIEOFFSET=5 3D=5>
<PARAMS FONT=2 SIZE=10>
</CHART>
```

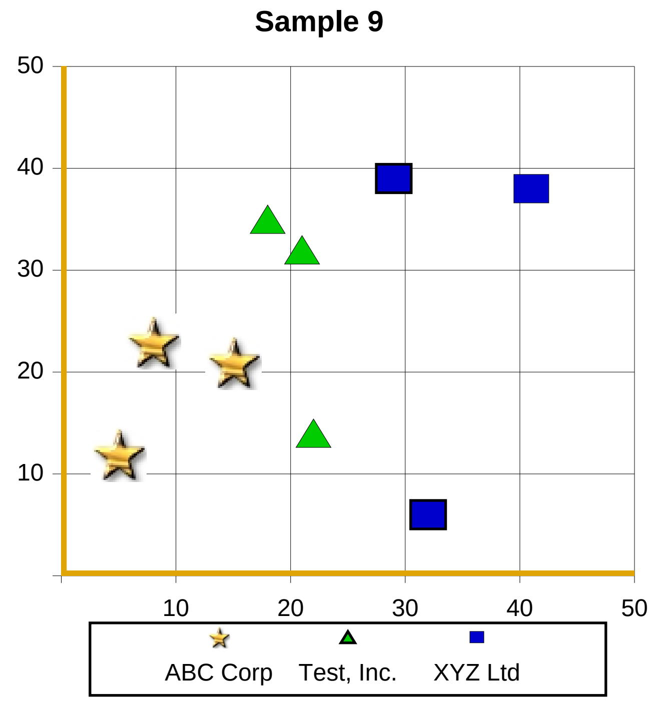


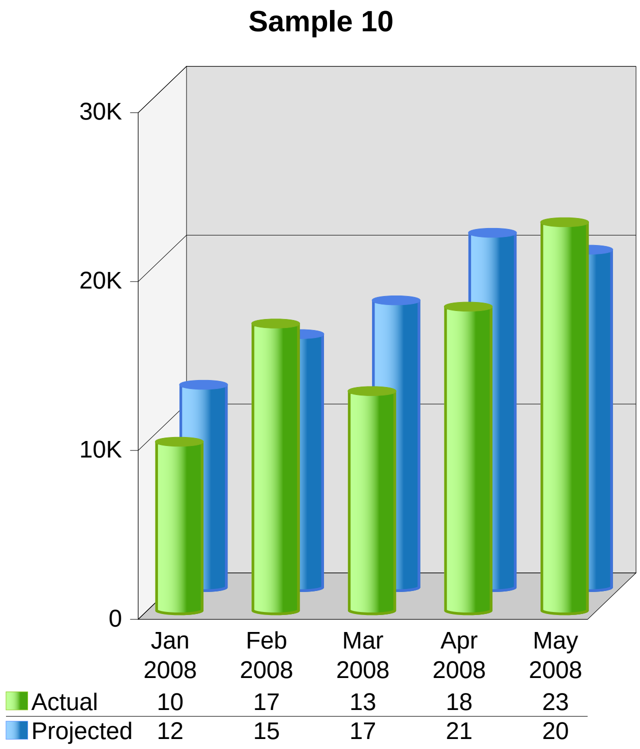
```
<SHADING NAME="gw" COLOR1=#CCCCCC COLOR2=#FFFFFF
COLORARY="0,.2,0,1">
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART WIDTH=35 HEIGHT=35>
<PARAMS TYPE="AREA" BEZIER>
<PARAMS TITLE="Sample 7" TITLEFONT=5 TITLESIZE=11>
<PARAMS 3D=15 3DBACKSHADE="gw">
<DATALINE LINE=1 VALUES=5,12,15,21,8
SCOLOR="#20710a">
<DATALINE LINE=2 VALUES=21,32,18,35,22
SCOLOR="#f5b903">
<DATALINE LINE=3 VALUES=41,38,29,39,32
SCOLOR="#820041">
<PARAMS YSCALE=0,50>
<PARAMS YLABEL="|10|20|30|40|50">
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=2 SIZE=9>
<PARAMS LEGEND="ABC Corp|Test, Inc.|XYZ Ltd">
</CHART>
```



```
<SHADING NAME="gw" COLOR1=#CCCCCC COLOR2=#FFFFFF
COLORARY="0,.2,0,1">
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=40 X2=75 HEIGHT=35>
<PARAMS TYPE="BAR-LINE" BEZIER>
<PARAMS TITLE="Sample 8" TITLEFONT=5 TITLESIZE=11>
<DATALINE LINE=1 VALUES=5,12,15,21,8
NODESCOLOR=blue NODEFCOLOR=yellow WIDTH=2>
<DATALINE LINE=2 VALUES=,,18,35,22 NODESIZE=3
WIDTH=2>
<DATALINE LINE=3 VALUES=41,38,29,39,32
NODESCOLOR=black WIDTH=2>
<PARAMS YSCALE=0,50 3DBACKSHADE="gw">
<PARAMS YGRID=5>
<PARAMS XGRID=4>
<PARAMS GRIDDASH=3,3>
<PARAMS LINENODE NODEWIDTH=1>
<PARAMS YLABEL="|10|20|30|40|50">
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=2 SIZE=9>
<PARAMS LEGEND="ABC Corp|Test, Inc.|XYZ Ltd">
</CHART>
```

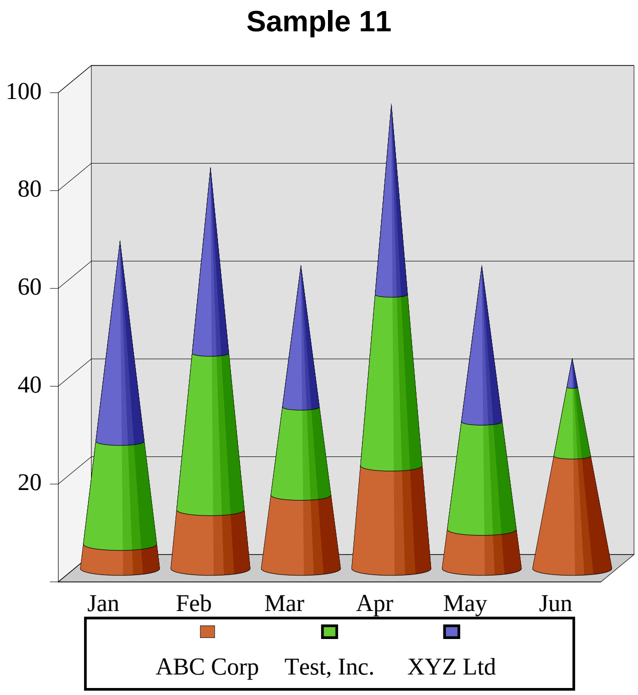
Charts



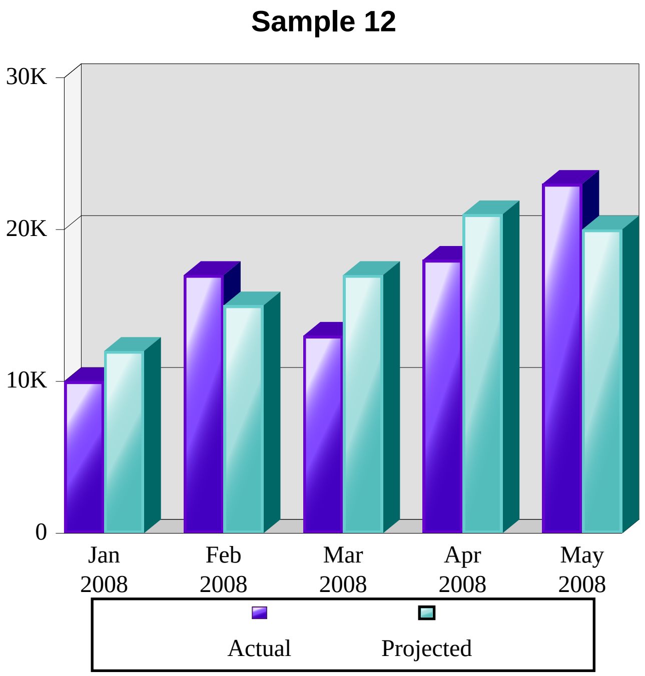


```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART WIDTH=35 HEIGHT=35>
<PARAMS TYPE="SCATTER">
<PARAMS TITLE="Sample 9" TITLEFONT=5 TITLESIZE=11>
<DATALINE LINE=1 VALUES=5,12,15,21,8,23
NODE="smallstr.jpg">
<DATALINE LINE=2 VALUES=21,32,18,35,22,14 NODE=7>
<DATALINE LINE=3 VALUES=41,38,29,39,32,6>
<PARAMS YSCALE=0,50>
<PARAMS XSCALE=0,50>
<PARAMS XGRID=5>
<PARAMS YGRID=5>
<PARAMS BORDEROPTS="border-left:4 solid
#e0a404;border-bottom:4 solid #e0a404">
<PARAMS NODESIZE=5 NODESCOLOR=black>
<PARAMS YLABEL="|10|20|30|40|50">
<PARAMS XLABEL="|10|20|30|40|50">
<PARAMS FONT=2 SIZE=9>
<PARAMS LEGEND="ABC Corp|Test, Inc.|XYZ Ltd">
</CHART>
```

```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<SHADING NAME="wb" COLOR1=#95d1ff COLOR2=#1774ba
COLORARY="0,0,.65,0">
<SHADING NAME="wg" COLOR1=#bdff94 COLOR2=#46a50b
COLORARY="0,0,.65,0">
<CHART X1=40 X2=75 HEIGHT=35>
<PARAMS TYPE="BAR-LINE" BGCOLOR=#E0E0E0>
<PARAMS 3D=30 BARSTYLE=Cylinder BARCOLUMNS>
<PARAMS TITLE="Sample 10" TITLEFONT=5
TITLESIZE=11>
<BARDATAMULT VALUES=10,17,13,18,23
LABELS=10|17|13|18|23 FCOLOR=#99CC33 LEGEND=Actual
SHADING=wg>
<BARDATAMULT VALUES=12,15,17,21,20
LABELS=12|15|17|21|20 FCOLOR=#6699FF
LEGEND=Projected SHADING=wb>
<BARBORDER WIDTH=2 STRENGTH=-15>
<PARAMS YSCALE=0,30>
<PARAMS YGRID=3 3DTOP=-10 3DSIDE=30 REVEFFECT>
<PARAMS BARSPACE=50>
<PARAMS YLABEL="0|10K|20K|30K">
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=2 SIZE=9>
</CHART>
```

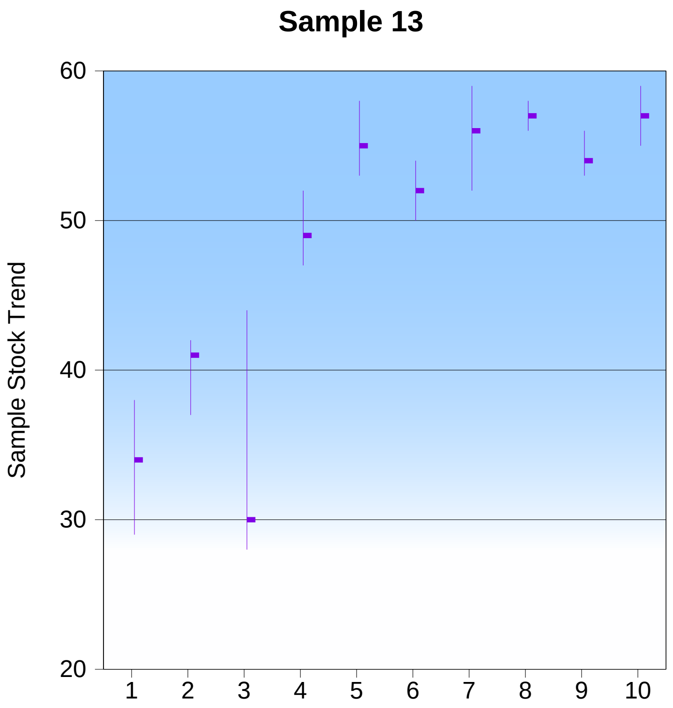


```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART WIDTH=35 HEIGHT=35>
<PARAMS TYPE="BAR-LINE" BARSTYLE=Cone>
<PARAMS TITLE="Sample 11" TITLEFONT=5
TITLESIZE=11>
<BARSERIES VALUES=5,12,15,21,8,23 FCOLOR=#CC6633>
<BARSERIES VALUES=21,32,18,35,22,14
FCOLOR=#66CC33>
<BARSERIES VALUES=41,38,29,39,32,6 FCOLOR=#6666CC>
<PARAMS BARSPACE=12 BGCOLOR=#E0E0E0>
<PARAMS 3D=20>
<PARAMS YSCALE=0,100>
<PARAMS YGRID=5>
<PARAMS YLABEL="|20|40|60|80|100">
<PARAMS XLABEL="Jan|Feb|Mar|Apr|May|Jun">
<PARAMS FONT=3 SIZE=9>
<PARAMS LEGEND="ABC Corp|Test, Inc.|XYZ Ltd">
</CHART>
```

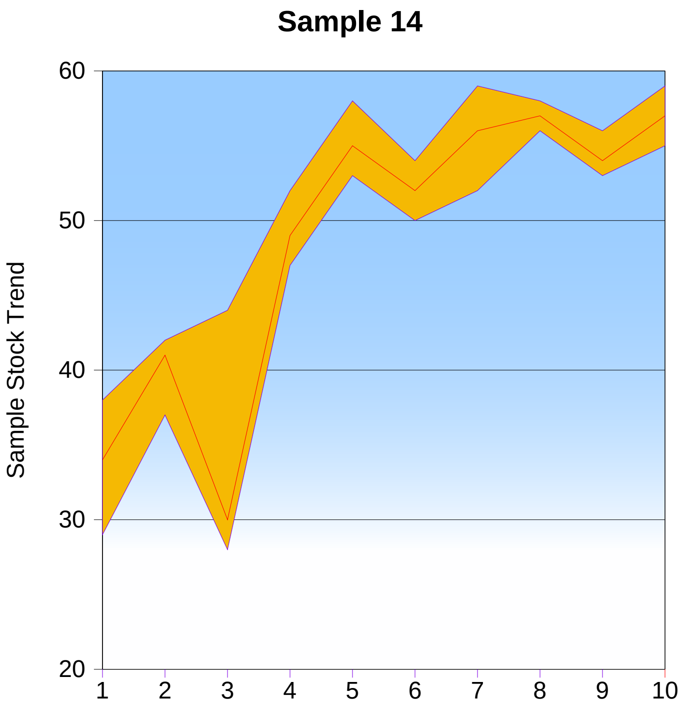


```
<SHADING NAME="a" COLOR1=#4400c0 COLOR2=#8149ff
COLOR3=#e9e1ff COLORARY=".3,.7,0,0">
<SHADING NAME="p" COLOR1=#53bdbc COLOR2=#a3dddc
COLOR3=#e4f6f6 COLORARY=".3,.7,0,0">
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=40 X2=75 HEIGHT=35>
<PARAMS TYPE="BAR-LINE" BGCOLOR=#E0E0E0>
<PARAMS 3D=10 3DTOP=-20 3DSIDE=-50>
<BARBORDER WIDTH=2 STRENGTH=-10>
<PARAMS TITLE="Sample 12" TITLEFONT=5
TITLESIZE=11>
<PARAMS BARDATA=10,12,17,15,13,17,18,21,23,20>
<PARAMS DATAMULT=2>
<PARAMS LEGEND="Actual|Projected">
<PARAMS YSCALE=0,30>
<PARAMS YGRID=3>
<PARAMS BARSHADING=a,p,a,p,a,p,a,p,a,p>
<PARAMS BARCOLOR=11,13,11,13,11,13,11,13,11,13>
<PARAMS YLABEL="0|10K|20K|30K">
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=3 SIZE=9>
</CHART>
```

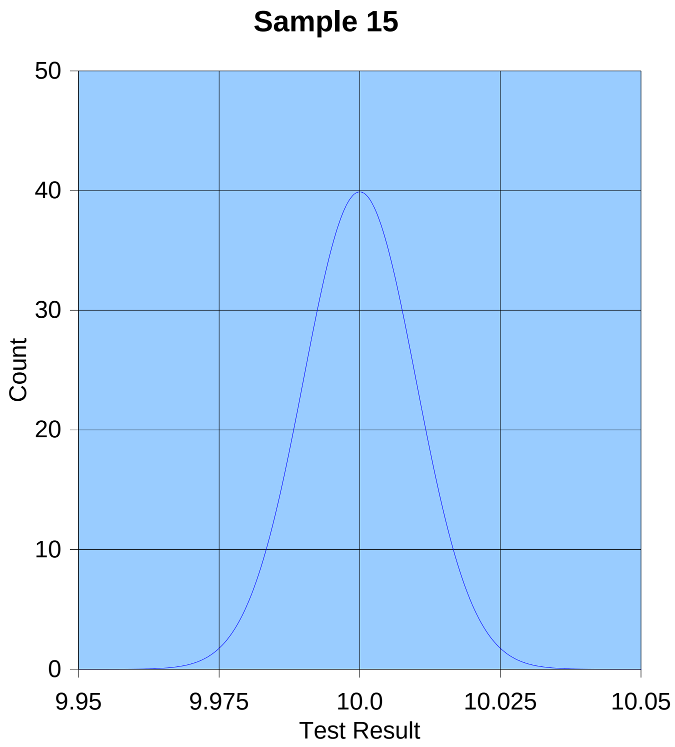
Charts



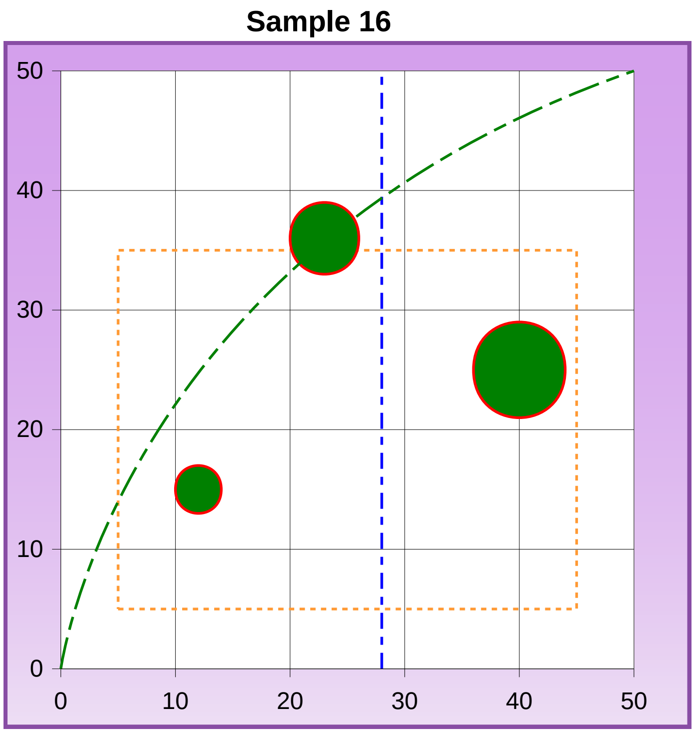
```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<SHADING NAME="bw" COLOR1=#99CCFF COLOR2=#FFFFFF
COLORARY="0,.2,0,1"<
<CHART WIDTH=35 HEIGHT=35>
<PARAMS TYPE="STOCK">
<PARAMS TITLE="Sample 13" TITLEFONT=5
TITLESIZE=11>
<PARAMS HIGH=38,42,44,52,58,54,59,58,56,59>
<PARAMS LOW=29,37,28,47,53,50,52,56,53,55>
<PARAMS AVG=34,41,30,49,55,52,56,57,54,57>
<PARAMS YSCALE=20,60>
<PARAMS YLABEL="20|30|40|50|60">
<PARAMS YGRID=4 3DBACKSHADE="bw">
<PARAMS XGRID=1 SHOWXTICKS=On>
<PARAMS BARSPACE=10>
<PARAMS ALLBARCOLOR=11>
<PARAMS YAXISLABEL="Sample Stock Trend">
<PARAMS YAXISOFFSET=5>
<PARAMS XLABEL="1|2|3|4|5|6|7|8|9|10">
<PARAMS FONT=2 SIZE=9>
</CHART>
```



```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<SHADING NAME="bw" COLOR1=#99CCFF COLOR2=#FFFFFF
COLORARY="0,.2,0,1"<
<CHART WIDTH=35 HEIGHT=35>
<PARAMS TYPE="STOCK">
<PARAMS TITLE="Sample 14" TITLEFONT=5
TITLESIZE=11>
<PARAMS HIGH=38,42,44,52,58,54,59,58,56,59>
<PARAMS LOW=29,37,28,47,53,50,52,56,53,55>
<PARAMS AVG=34,41,30,49,55,52,56,57,54,57>
<PARAMS YSCALE=20,60>
<PARAMS YLABEL="20|30|40|50|60">
<PARAMS YGRID=4 3DBACKSHADE="bw">
<PARAMS XGRID=1 SHOWXTICKS=On>
<PARAMS STOCKSHADE="#f5b903 NODESCOLOR="#ff0000"">
<PARAMS ALLBARCOLOR=11>
<PARAMS YAXISLABEL="Sample Stock Trend">
<PARAMS YAXISOFFSET=5>
<PARAMS XLABEL="1|2|3|4|5|6|7|8|9|10">
<PARAMS FONT=2 SIZE=9>
</CHART>
```

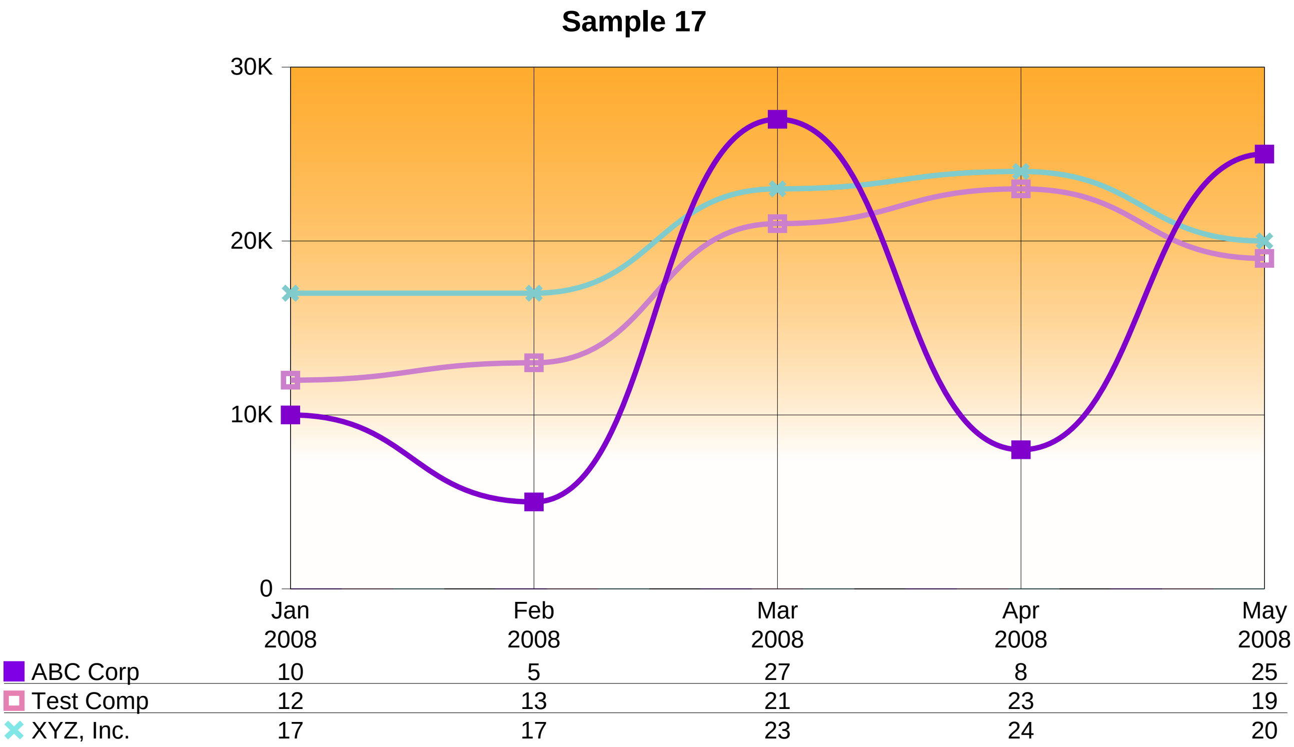


```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART WIDTH=35 HEIGHT=35>
<PARAMS TYPE="BAR-LINE">
<PARAMS TITLE="Sample 15" TITLEFONT=5
TITLESIZE=11>
<CHARTNORMDIST X1=9.95 X2=10.05 MU=10 SIGMA=.01
WIDTH=.1 SCOLOR=blue>
<PARAMS YSCALE=0,50>
<PARAMS XSCALE=9.95,10.05>
<PARAMS XGRID=4>
<PARAMS YGRID=5>
<PARAMS NODESIZE=5 BGCOLOR=#99CCFF>
<PARAMS YLABEL="0|10|20|30|40|50">
<PARAMS XLABEL="9.95|9.975|10.0|10.025|10.05">
<PARAMS XAXISLABEL="Test Result">
<PARAMS YAXISOFFSET=2 XAXISOFFSET=1>
<PARAMS YAXISLABEL="Count">
<PARAMS FONT=2 SIZE=9>
</CHART>
```



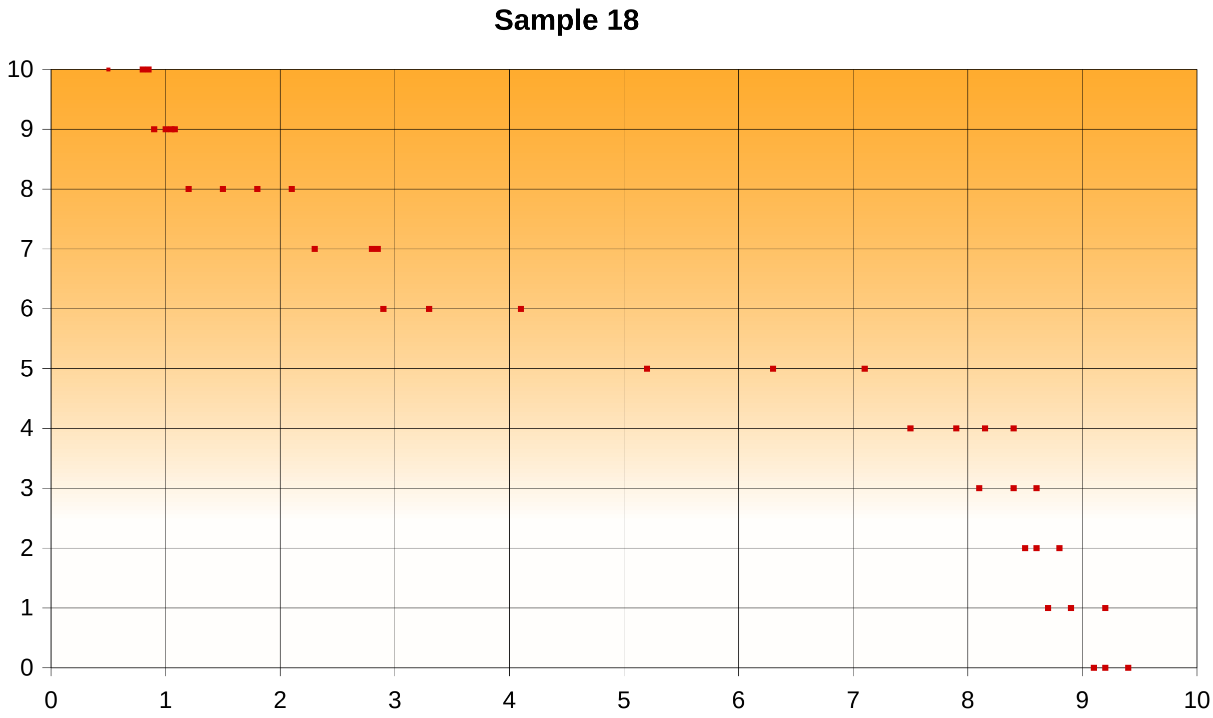
```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<SHADING NAME="pw" COLOR1=#d39dec COLOR2=#eee0f4
COLORARY="0,0,0,1.5">
<CHART X1=40 X2=75 HEIGHT=35>
<PARAMS TYPE="BAR-LINE">
<PARAMS TITLE="Sample 16" TITLEFONT=5
TITLESIZE=11>
<PARAMS XSCALE=0,50 YSCALE=0,50>
<PARAMS YGRID=5 XGRID=5>
<CHARTCIRCLE X=12 Y=15 RADIUS=2 FCOLOR=green
SCOLOR=red FILL>
<CHARTCIRCLE X=23 Y=36 RADIUS=3 FCOLOR=green
SCOLOR=red FILL>
<CHARTCIRCLE X=40 Y=25 RADIUS=4 FCOLOR=green
SCOLOR=red FILL>
<CHARTLINE X1=28 Y1=0 X2=28 Y2=50 DASH=6,3,3,3
FCOLOR=white SCOLOR=blue>
<CHARTRECT X1=5 Y1=5 X2=45 Y2=35 ON=2 OFF=2
SCOLOR=1,.6,.2>
<CHARTCURVE X1=0 Y1=0 X2=50 Y2=50 CPX1=5 CPY1=35
DASH=12,3,5,3 FCOLOR=white SCOLOR=green>
<PARAMS YLABEL="0|10|20|30|40|50">
<PARAMS XLABEL="0|10|20|30|40|50">
<PARAMS FONT=2 SIZE=9 BORDEROPTS="padding:5 10 10
10;shade:pw;border:3 solid #884da4" BGCOLOR=#FFF>
</CHART>
```

Charts

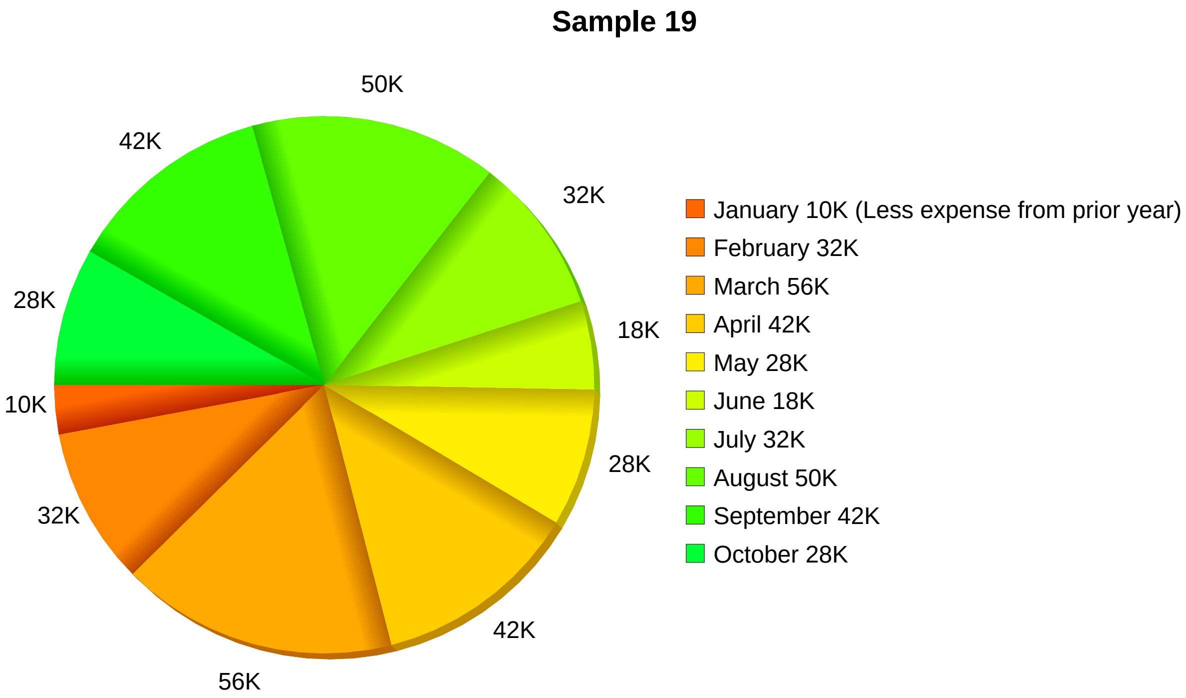

Sample 17
30K
20K
10K
0
Jan
2008
Feb
2008
Mar
2008
Apr
2008
May
2008
■ ABC Corp 10 5 27 8 25
□ Test Comp 12 13 21 23 19
✕ XYZ, Inc. 17 17 23 24 20

```
<SHADING NAME="orange" COLOR1=#FF9900 COLOR2=#FFFFFF COLORARY="0,.25,0,2">
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=5 X2=75 HEIGHT=35>
<PARAMS TYPE="BAR-LINE" BEZIER>
<PARAMS LINELAB LINENODE NODEWIDTH=2>
<PARAMS TITLE="Sample 17" TITLEFONT=5 TITLESIZE=11>
<DATALINE LINE=1 VALUES=10,5,27,8,25 LABELS="10|5|27|8|25" WIDTH=2>
<DATALINE LINE=2 VALUES=12,13,21,23,19 LABELS="12|13|21|23|19" WIDTH=2>
<DATALINE LINE=3 VALUES=17,17,23,24,20 LABELS="17|17|23|24|20" WIDTH=2>
<PARAMS DATAMULT=3>
<PARAMS DATAMLAB="ABC Corp|Test Comp|XYZ, Inc.">
<PARAMS YSCALE=0,30>
<PARAMS YGRID=3>
<PARAMS XGRID=4>
<PARAMS 3DBACKSHADE="orange">
<PARAMS BARCOLOR=11,14,13>
<PARAMS LINECOLOR=11,14,13>
<PARAMS YLABEL="0|10K|20K|30K">
<PARAMS XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|May\n2008">
<PARAMS FONT=2 SIZE=9>
</CHART>
```

Charts

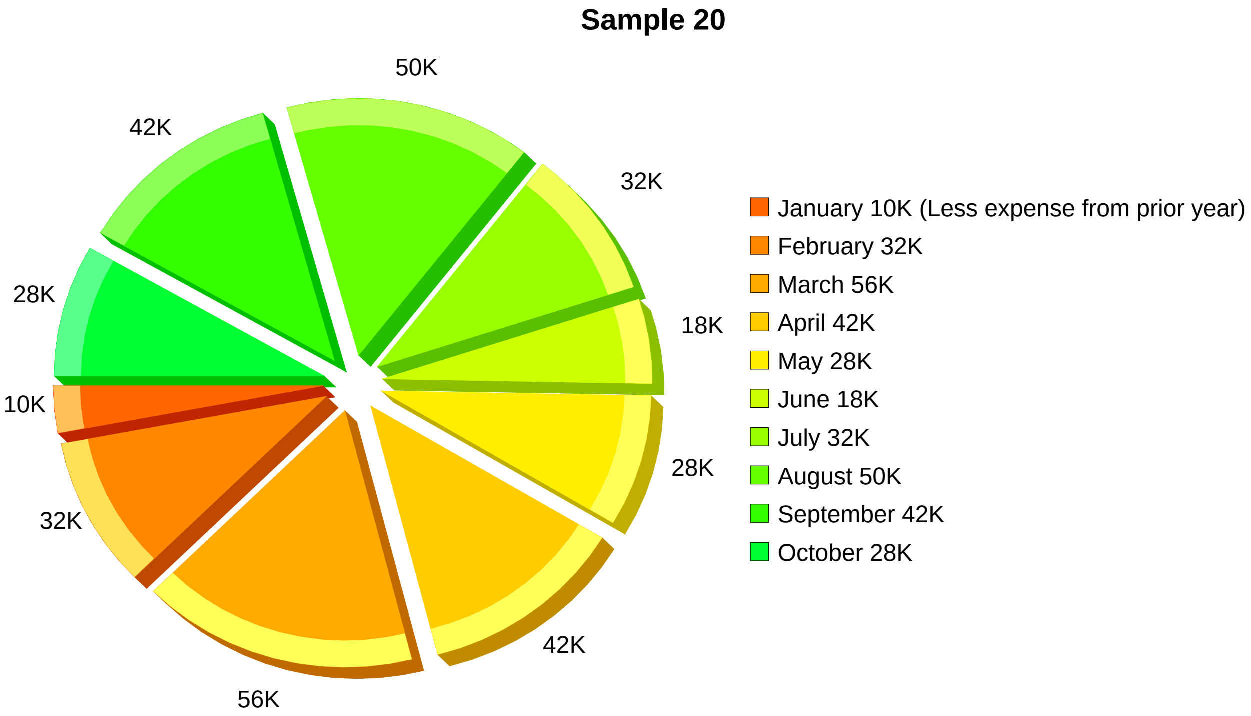


```
<SHADING NAME="orange" COLOR1=#FF9900 COLOR2=#FFFFFF COLORARY="0,.25,0,2">
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=5 X2=75 HEIGHT=35>
<PARAMS TYPE="SCATTER">
<PARAMS TITLE="Sample 18" TITLEFONT=5 TITLESIZE=11>
<CHARTPOINTS VAR=X POINTS=M WIDTH=1 COLOR=red NODESIZE=.5>
.5,10,.8,10,.85,10
.9,9,1,9,1.05,9,1.08,9
1.2,8,1.5,8,1.8,8,2.1,8
2.3,7,2.8,7,2.85,7
2.9,6,3.3,6,4.1,6
5.2,5,6.3,5,7.1,5
7.5,4,7.9,4,8.15,4,8.4,4
8.1,3,8.4,3,8.6,3
8.5,2,8.6,2,8.8,2
8.7,1,8.9,1,9.2,1
9.1,0,9.2,0,9.4,0
</CHARTPOINTS>
<PARAMS XSCALE=0,10> <PARAMS YSCALE=0,10>
<PARAMS YGRID=10> <PARAMS XGRID=10>
<PARAMS 3DBACKSHADE="orange">
<PARAMS YLABEL="0|1|2|3|4|5|6|7|8|9|10">
<PARAMS XLABEL="0|1|2|3|4|5|6|7|8|9|10">
<PARAMS FONT=2 SIZE=9>
</CHART>
```

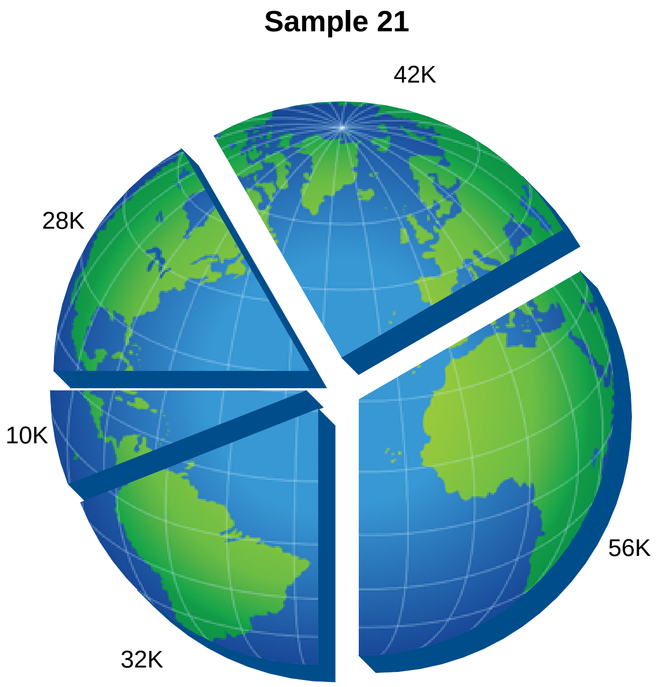


```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=5 X2=75 HEIGHT=35>
<PARAMS TYPE="PIE">
<PARAMS TITLE="Sample 19" TITLEFONT=5 TITLESIZE=11>
<PARAMS BARDATA=10,32,56,42,28,18,32,50,42,28>
<PARAMS DATALAB="10K|32K|56K|42K|28K|18K|32K|50K|42K|28K">
<PARAMS LEGEND="January 10K (Less expense from prior year)|February 32K|March 56K|April 42K|May
28K|June 18K|July 32K|August 50K|September 42K|October 28K">
<PARAMS 3D=1 OUTSIDELABELS>
<SLICEBORDER SIDES="2" STRENGTH=-25 WIDTH=20 SHADE>
<SLICECOLOR VALUE="#f60">
<SLICECOLOR VALUE="#f80">
<SLICECOLOR VALUE="#fa0">
<SLICECOLOR VALUE="#fc0">
<SLICECOLOR VALUE="#fe0">
<SLICECOLOR VALUE="#cf0">
<SLICECOLOR VALUE="#9f0">
<SLICECOLOR VALUE="#6f0">
<SLICECOLOR VALUE="#3f0">
<SLICECOLOR VALUE="#0f3">
<PARAMS FONT=2 SIZE=9>
</CHART>
```

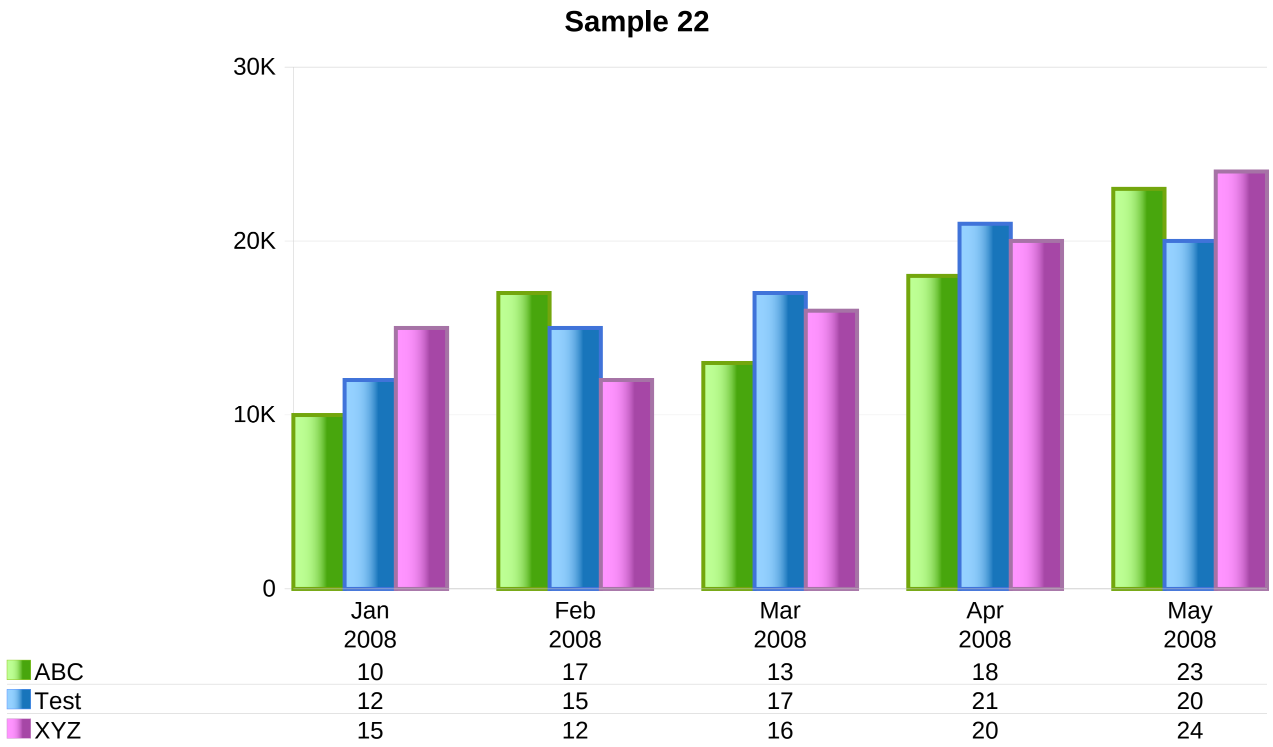
Charts



```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=5 X2=75 HEIGHT=35>
<PARAMS TYPE="PIE">
<PARAMS TITLE="Sample 20" TITLEFONT=5 TITLESIZE=11>
<PARAMS 3D=2>
<PARAMS BARDATA=10,32,56,42,28,18,32,50,42,28>
<PARAMS DATALAB="10K|32K|56K|42K|28K|18K|32K|50K|42K|28K">
<PARAMS LEGEND="January 10K (Less expense from prior year)|February 32K|March 56K|April 42K|May
28K|June 18K|July 32K|August 50K|September 42K|October 28K">
<PARAMS PIEOFFSET=5 OUTSIDELABELS>
<SLICEBORDER ARC STRENGTH=35 WIDTH=20<
<SLICECOLOR VALUE="#f60">
<SLICECOLOR VALUE="#f80">
<SLICECOLOR VALUE="#fa0">
<SLICECOLOR VALUE="#fc0">
<SLICECOLOR VALUE="#fe0">
<SLICECOLOR VALUE="#cf0">
<SLICECOLOR VALUE="#9f0">
<SLICECOLOR VALUE="#6f0">
<SLICECOLOR VALUE="#3f0">
<SLICECOLOR VALUE="#0f3">
<PARAMS FONT=2 SIZE=9>
</CHART>
```


Sample 21
42K
28K
10K
32K
56K

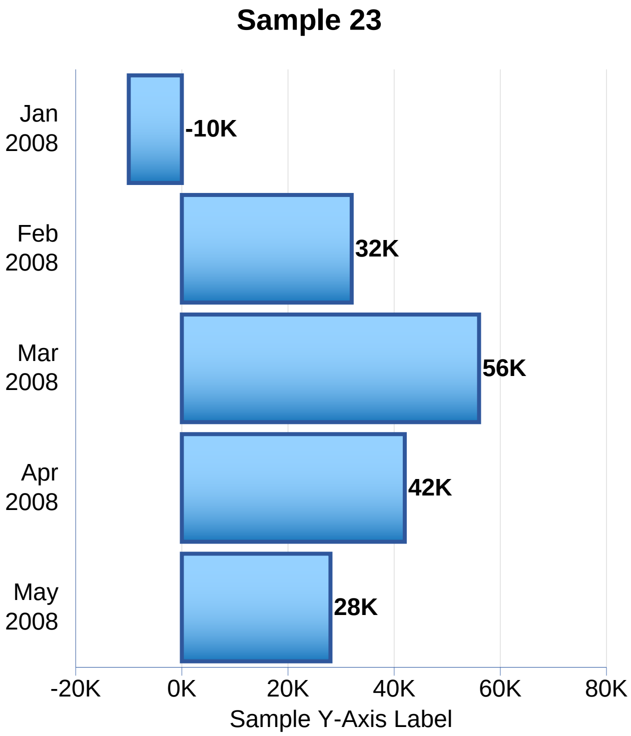
```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=5 X2=40 HEIGHT=35>
<PARAMS TYPE="PIE" 3d=3>
<PARAMS TITLE="Sample 21" TITLEFONT=5 TITLESIZE=11>
<SLICECOLOR VALUE=#338dcc 3DSIDE=-30>
<PARAMS IMAGE="globe.jpg" IMAGESCALE=65>
<PARAMS BARDATA=10,32,56,42,28>
<PARAMS DATALAB="10K|32K|56K|42K|28K">
<PARAMS PIEOFFSET=5 OUTSIDELABELS>
<PARAMS FONT=2 SIZE=9>
</CHART>
```

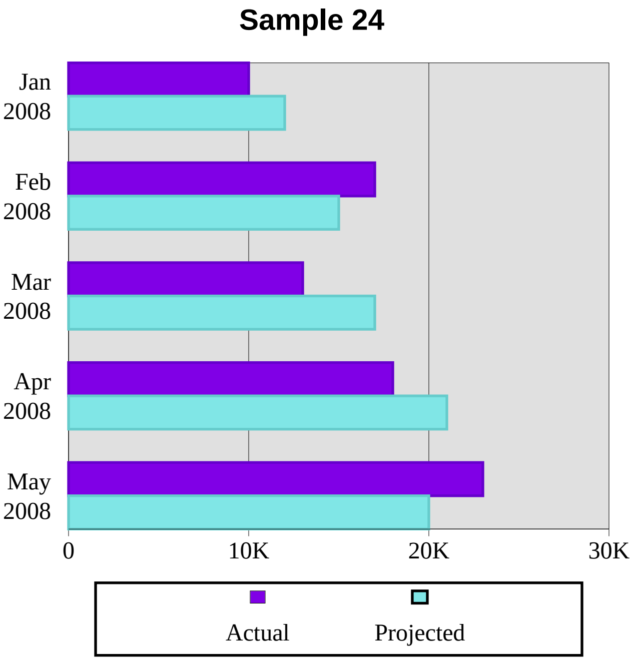


	Jan 2008	Feb 2008	Mar 2008	Apr 2008	May 2008
🟩 ABC	10	17	13	18	23
🟦 Test	12	15	17	21	20
🟪 XYZ	15	12	16	20	24

```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<SHADING NAME="wb" COLOR1=#95d1ff COLOR2=#1774ba COLORARY="0,0,.65,0">
<SHADING NAME="wg" COLOR1=#bdff94 COLOR2=#46a50b COLORARY="0,0,.65,0">
<SHADING NAME="wp" COLOR1=#ff94ff COLOR2=#a546a5 COLORARY="0,0,.65,0">
<CHART X1=5 X2=75 HEIGHT=35>
<PARAMS TYPE="BAR-LINE">
<PARAMS TITLE="Sample 22" TITLEFONT=5 TITLESIZE=11>
<BARDATAMULT VALUES=10,17,13,18,23 LABELS=10|17|13|18|23 FCOLOR=#99CC33 LEGEND=ABC Corp SHADING=wg>
<BARDATAMULT VALUES=12,15,17,21,20 LABELS=12|15|17|21|20 FCOLOR=#6699FF LEGEND=Test Co. SHADING=wb>
<BARDATAMULT VALUES=15,12,16,20,24 LABELS=15|12|16|20|24 FCOLOR=#CC99CC LEGEND=XYZ Inc. SHADING=wp>
<BARBORDER WIDTH=1.5 STRENGTH=-15>
<PARAMS YSCALE=0,30>
<PARAMS YGRID=3>
<PARAMS BASESCOLOR=#ccc GRIDSCOLOR=#ccc>
<PARAMS YLABEL="0|10K|20K|30K">
<PARAMS XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|May\n2008">
<PARAMS FONT=2 SIZE=9>
</CHART>
```

Charts

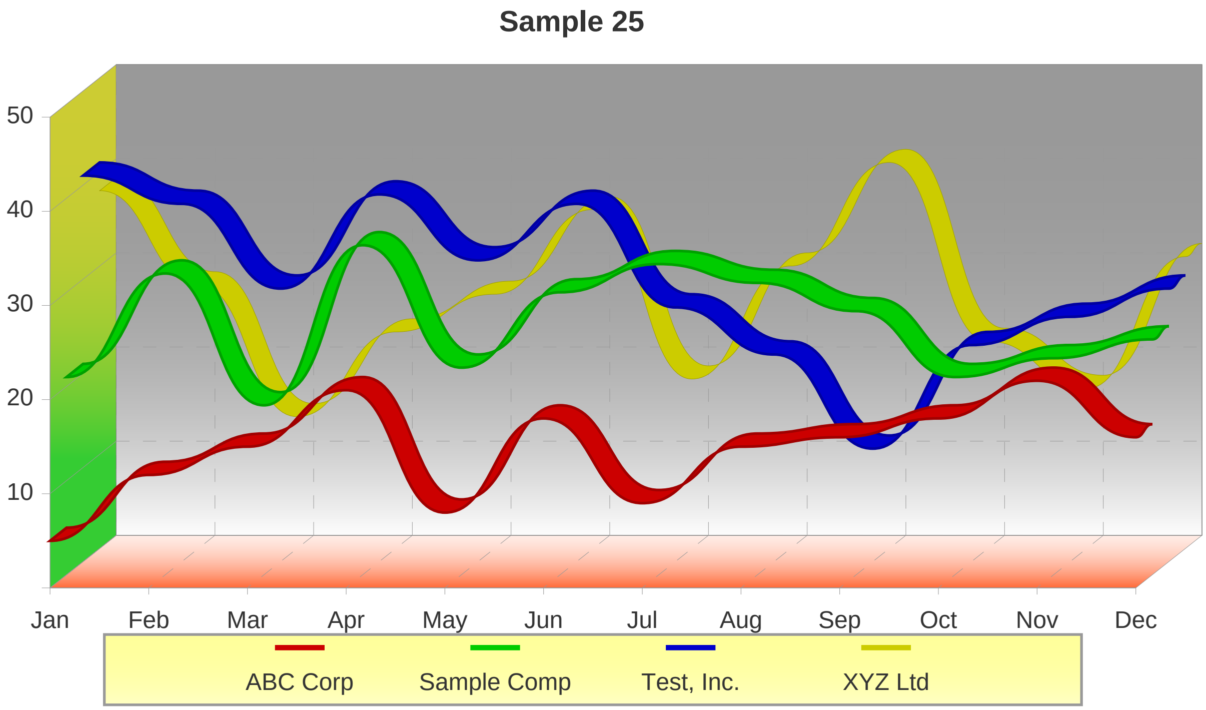




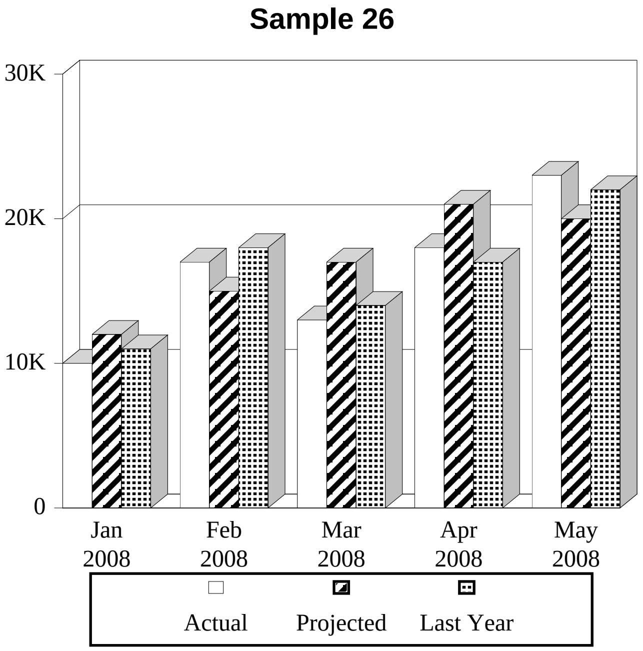
```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<SHADING NAME="wbh" COLOR1=#95d1ff COLOR2=#1774ba
COLORARY="0,0,0,1">
<CHART HEIGHT=35 WIDTH=30>
<PARAMS TYPE="BAR-LINE">
<PARAMS TITLE="Sample 23" TITLEFONT=5
TITLESIZE=11>
<BARBORDER WIDTH=1.5 STRENGTH=-5>
<PARAMS BARDATA=-10,32,56,42,28>
<PARAMS DATALAB="-10K|32K|56K|42K|28K">
<PARAMS YSCALE=-20,80 HORIZONTAL>
<PARAMS YLABEL="-20K|0K|20K|40K|60K|80K">
<PARAMS BARSPACE=10 BASESCOLOR=#3c64a9
GRIDSCOLOR=#ccc>
<PARAMS YGRID=5 ALLBARSHADING=wbh>
<PARAMS ALLBARCOLOR=#3c64a9 LABFONT=5
XLABELFONT=2>
<PARAMS YAXISLABEL="Sample Y-Axis Label">
<PARAMS YAXISOFFSET=6>
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=2 SIZE=9>
```

```
<SHADING NAME="a" COLOR1=#4400c0 COLOR2=#8149ff
COLOR3=#e9e1ff COLORARY=".3,.7,0,0">
<SHADING NAME="p" COLOR1=#53bdbc COLOR2=#a3dddc
COLOR3=#e4f6f6 COLORARY=".3,.7,0,0">
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=40 X2=75 HEIGHT=35>
<PARAMS TYPE="BAR-LINE" BGCOLOR=#E0E0E0>
<BARBORDER WIDTH=1 STRENGTH=-10>
<PARAMS TITLE="Sample 24" TITLEFONT=5
TITLESIZE=11>
<PARAMS BARDATA=10,12,17,15,13,17,18,21,23,20>
<PARAMS DATAMULT=2>
<PARAMS LEGEND="Actual|Projected" LEGENDOFFSET=-
10>
<PARAMS YSCALE=0,30>
<PARAMS YGRID=3 XGRID=1 HORIZONTAL>
<PARAMS BARCOLOR=11,13,11,13,11,13,11,13,11,13>
<PARAMS YLABEL="0|10K|20K|30K">
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=3 SIZE=9>
</CHART>
```

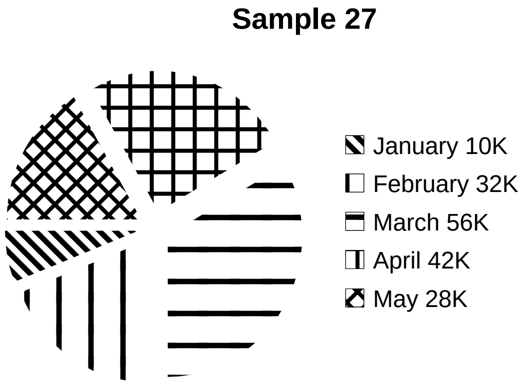
Charts


Sample 25
50
40
30
20
10
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
ABC Corp
Sample Comp
Test, Inc.
XYZ Ltd

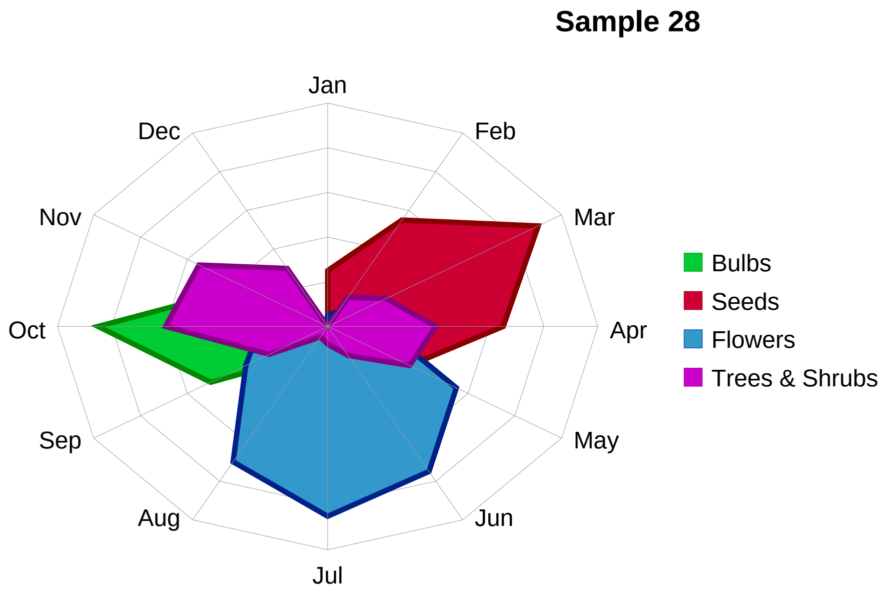
```
<SHADING NAME="yellgreen" COLOR1=#CCCC33 COLOR2=#33CC33 COLORARY="0,.25,0,1">
<SHADING NAME="whitered" COLOR1=#FFFFFF COLOR2=#FF6633 COLORARY="0,0,0,2">
<SHADING NAME="greywhite" COLOR1=#999999 COLOR2=#FFFFFF COLORARY="0,0,0,1">
<SHADING NAME="yellwhite" COLOR1=#FFFF99 COLOR2=#FFFFFF COLORARY="0,-.5,0,1.5">
<CHART WIDTH=70 HEIGHT=35>
<PARAMS TYPE="RIBBON" BEZIER>
<PARAMS TITLE="Sample 25" TITLEFONT=5 TITLESIZE=11>
<PARAMS 3D=40>
<DATALINE LINE=1 VALUES=5,12,15,21,8,18,9,15,16,18,22,16>
<DATALINE LINE=2 VALUES=21,32,18,35,22,30,33,31,28,21,23,25>
<DATALINE LINE=3 VALUES=41,38,29,39,32,38,27,22,12,23,26,29>
<DATALINE LINE=4 VALUES=38,28,14,23,27,36,18,30,41,22,17,31>
<PARAMS YSCALE=0,50>
<PARAMS YLABEL="|10|20|30|40|50">
<PARAMS BASEFCOLOR=#333333 BASESCOLOR=#999999>
<PARAMS GRIDON=5 GRIDOFF=5 GRIDFCOLOR="#CCCCCC">
<PARAMS 3DLEFTSHADE=yellgreen>
<PARAMS 3DBOTTOMSHADE=whitered>
<PARAMS 3DBACKSHADE=greywhite>
<PARAMS LEGENDSHADE=yellwhite>
<PARAMS XGRID=11 YGRID=5>
<PARAMS XLABEL="Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec">
<PARAMS FONT=2 SIZE=9>
<PARAMS LEGEND="ABC Corp|Sample Comp|Test, Inc.|XYZ Ltd">
</CHART>
```



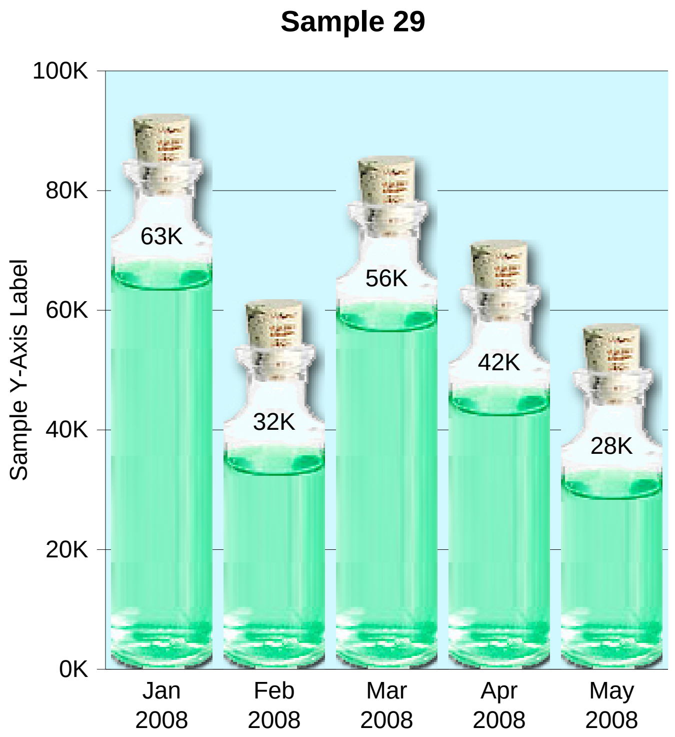
```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART WIDTH=35 HEIGHT=35>
<CHARTCOLOR COLORNUM=14 COLOR=white>
<PARAMS TYPE="BAR-LINE">
<PARAMS 3D=10>
<PARAMS TITLE="Sample 26" TITLEFONT=5
TITLESIZE=11>
<PARAMS
BARDATA=10,12,11,17,15,18,13,17,14,18,21,17,23,20
,22>
<PARAMS BARPATTERN=0,1,8,0,1,8,0,1,8,0,1,8,0,1,8>
<PARAMS DATAMULT=3>
<PARAMS LEGEND="Actual|Projected|Last Year">
<PARAMS YSCALE=0,30>
<PARAMS YGRID=3>
<PARAMS ALLBARCOLOR=14>
<PARAMS YLABEL="0|10K|20K|30K">
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=3 SIZE=9>
</CHART>
```



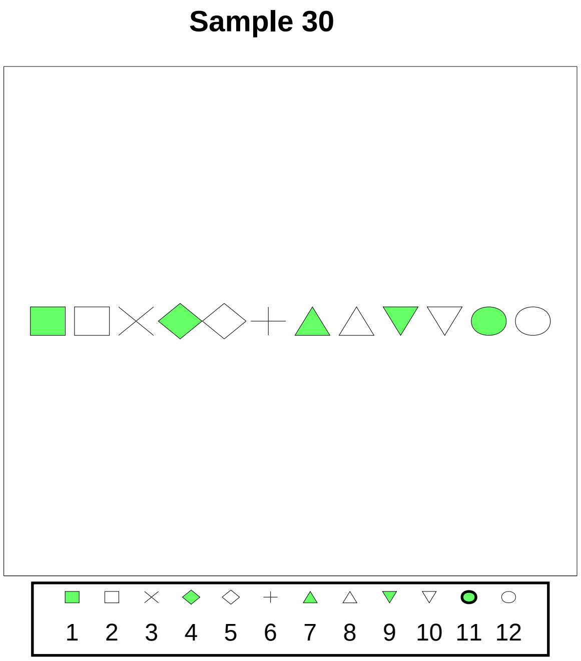
```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=40 X2=75 HEIGHT=20>
<CHARTCOLOR COLORNUM=14 COLOR=white>
<PARAMS TYPE="PIE">
<PARAMS TITLE="Sample 27" TITLEFONT=5
TITLESIZE=11>
<PARAMS BARDATA=10,32,56,42,28>
<PARAMS BARPATTERN=2,3,4,5,6>
<PARAMS LEGEND="Jan 10K|Feb 32K|Mar 56K|Apr
42K|May 28K">
<PARAMS PIEOFFSET=5>
<PARAMS ALLBARCOLOR=14>
<PARAMS FONT=2 SIZE=9>
</CHART>
```



```
<CHART WIDTH=70 HEIGHT=35>
<PARAMS TYPE="RADAR">
<PARAMS TITLE="Sample 28" TITLEFONT=5 TITLESIZE=11>
<PARAMS YGRID=3 GRIDSCOLOR=#999>
<PARAMS XLABEL="Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec">
<RADARSERIES VALUES=0,0,0,0,0,0,0,1500,5000,8500,3500,500 LEGEND="Bulbs"
   FCOLOR="#00CC33" SCOLOR="#008800" FILL WIDTH=2>
<RADARSERIES VALUES=2500,5500,9000,6500,3500,0,0,0,0,0,0,0 LEGEND="Seeds"
   FCOLOR="#CC0033" SCOLOR="#88000" FILL WIDTH=2>
<RADARSERIES VALUES=500,750,1500,2000,5500,7500,8500,7000,3500,2500,500,100 LEGEND="Flowers"
   FCOLOR="#3399CC" SCOLOR="#002288" FILL WIDTH=2>
<RADARSERIES VALUES=0,1500,2500,4000,3500,1500,800,550,2500,6000,5500,3000 LEGEND="Trees & Shrubs"
   FCOLOR="#CC00CC" SCOLOR="#880088" FILL WIDTH=2>
<PARAMS FONT=2 SIZE=9>
</CHART>
```



```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<CHART X1=40 X2=75 HEIGHT=35>
<PARAMS TYPE="BAR-LINE">
<PARAMS TITLE="Sample 29" TITLEFONT=5
TITLESIZE=11>
<PARAMS BARDATA=63,32,56,42,28>
<PARAMS DATALAB="63K|32K|56K|42K|28K">
<PARAMS YSCALE=0,100 BGCOLOR="#d1f8fe">
<PARAMS YLABEL="0K|20K|40K|60K|80K|100K">
<PARAMS BARSPACE=10>
<PARAMS YGRID=5 LEGENDOFFSET=5>
<PARAMS IMAGE="chmid.jpg">
<PARAMS IMAGETOP="chtop.jpg">
<PARAMS IMAGEBOT="chbot.jpg">
<PARAMS IMAGETOPVALIGN="bottom">
<PARAMS YAXISLABEL="Sample Y-Axis Label">
<PARAMS YAXISOFFSET=5>
<PARAMS
XLABEL="Jan\n2008|Feb\n2008|Mar\n2008|Apr\n2008|M
ay\n2008">
<PARAMS FONT=2 SIZE=9>
</CHART>
```



Sample showing the various node types.

Gantt Charts

A Gantt chart is used to illustrate a project schedule. A table is used as the structure to setup and display a Gantt chart in Report Writer. The base parameters are specified in a table TD tag in the table header and each task is setup using a GANTT tag in the table rows.

Gantt Charts

Use the following options in the TD tag of the table header (TH row section) for the column in which you want to display the Gantt chart.

DATEFROM=text
DATETHRU=text
CURRTIME=text
CURRPOS=number
CURRCOLOR=color
WORKSPACE=number
CALTICKHEIGHT=number
CALHDRSIZE=number
CALDAYSIZE=number
DAYSTART=number
DAYSKIP=number
TARGETHEIGHT=number
BASEHEIGHT=number
PROGFCOLOR=color
PROGSCOLOR=color
TARGETFCOLOR=color
TARGETSCOLOR=color
BASELINEFCOLOR=color
BASELINESCOLOR=color

Parameter	Description
DATEFROM=text	The starting date/time for the chart. Enter as a string formatted as YYYY-MM-DD HH:Mi:SS. For example, "2015-10-15 13:30:00" for October 15, 2015 at 1:30pm.
DATETHRU=text	The ending date/time for the chart. Enter as a string formatted as YYYY-MM-DD HH:Mi:SS. For example, "2015-10-15 13:30:00" for October 15, 2015 at 1:30pm.
CURRTIME=text	The current date/time for the chart. Enter as a string formatted as YYYY-MM-DD HH:Mi:SS. For example, "2015-10-15 13:30:00" for October 15, 2015 at 1:30pm. This will be used to draw a vertical line through the chart at this time position.
CURRPOS=number	Use instead of CURRTIME. Specify the position based on the WORKSPACE value. This value should be from 1 to the value specified by WORKSPACE.

Gantt Charts

Parameter	Description
CURRCOLOR=color	The line color for the current date/time.
WORKSPACE=number	Optional. The total number of logical units for the length of time the chart represents. For example, if you set this value to 1000 then you may provide positional information from 1 to 1000 when drawing the chart.
WORKSPACE=number	Optional. The total number of units for the length of time the chart represents. For example, if you set this value to 1000 then you may provide positional information from 1 to 1000 when drawing the chart.
CALHDRSIZE=number	The font size of the calendar header for the month name.
CALDAYSIZE=number	The font size of the calendar header for the days.
DAYSTART=number	The starting day to draw (0 for the first, 1 for the second and so on).
DAYSKIP=number	The number of days to skip between labeling. Default is 0 to not skip any days.
TARGETHEIGHT=number	The height of the target bars in points.
BASEHEIGHT=number	The height of the base bars in points.
PROGFCOLOR=color	The fill color of the target bars to the left of the current date/time.
PROGSCOLOR=color	The stroke color of the target bars to the left of the current date/time.
TARGETFCOLOR=color	The fill color of the target bars to the right of the current date/time.
TARGETSCOLOR=color	The stroke color of the target bars to the right of the current date/time.
BASELINEFCOLOR=color	The fill color of the base bars.
BASELINESCOLOR=color	The stroke color of the base bars.

Gantt Charts

```
<GANTT
    PROG=text
    TARGET=text
    BASELINE=text
    BORDER=number>
```

Place this tag inside the TD tag that is in the same column as the chart setup. The GANTT tag can be though of as an image in the table cell that spans the entire cell. You may place any text above or below the GANTT tag.

Parameter	Description
PROG=text	A comma separated list of chart segments to draw using the PROGFCOLOR and PROGSCOLOR. Each entry has a 0- or 1- in the front to represent in-active (0) or active (1). Inactive time is drawn using a single line. Active time is drawn as a filled rectangle. The date/time may be specified either by a number (from 1 to the value specified by WORKSPACE) or a date/time in the form "YYYY-MM-DD HH:Mi:SS". Note that PROG and TARGET are appended together but using (optionally) different colors.
TARGET=text	A comma separated list of chart segments to draw using the TARGETFCOLOR and TARGETSCOLOR. Each entry has a 0- or 1- in the front to represent in-active (0) or active (1). Inactive time is drawn using a single line. Active time is drawn as a filled rectangle. The date/time may be specified either by a number (from 1 to the value specified by WORKSPACE) or a date/time in the form "YYYY-MM-DD HH:Mi:SS". Note that PROG and TARGET are appended together but using (optionally) different colors.

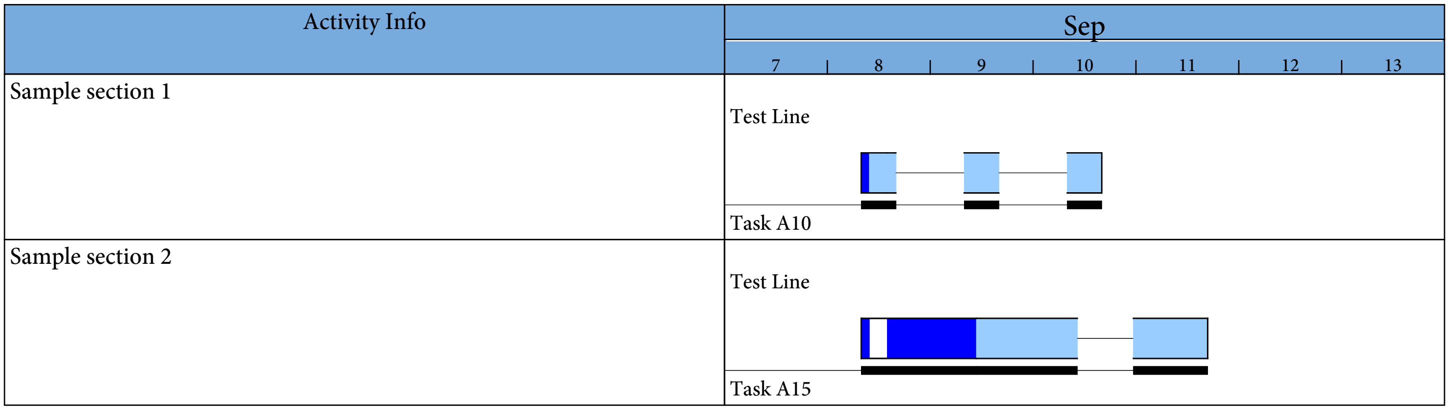
Gantt Charts

Parameter	Description
BASELINE=text	A comma separated list of chart segments to draw using the BASEFCOLOR and BASESCOLOR. Each entry has a 0- or 1- in the front to represent in-active (0) or active (1). Inactive time is drawn using a single line. Active time is drawn as a filled rectangle. The date/time may be specified either by a number (from 1 to the value specified by WORKSPACE) or a date/time in the form "YYYY-MM-DD HH:Mi:SS". The baseline is drawn below the PROG/TARGET section.
BORDER=number	Sets the border size around the target rectangle. A value of 1 is a line with a width of 1/72 of an inch.

Use the -allowbreaks option or setAllowBreaks method with the below example as some tags span lines.

```
<pdf>
<page>
<allowbreaks>
<table border=.5 size=9 forcewidth>
<th bgcolor='#77aadd'>
<td align="center" comptext=off nocomp>Activity Info</td>
<td width=250 comptext=off nocomp workspace=1000 currtime='2015-09-08 14:00:00' currcolor='#ff0000'
  datefrom='2015-09-07 00:00:00'
  datethru='2015-09-14 00:00:00'
  caltickheight=5 calhdrsize=11 caldaysize=7 targetheight=15 baseheight=3
  progfcolor="#0000ff"
  progscolor="#000"
  baselinefcolor="#000"
  baselinescolor="#000"
  targetfcolor="#99ccff"
  targetscolor="#000000" daystart=0 dayskip=0>
</td></th>
<tr>
<td>Sample section 1</td>
<td comptext=off nocomp size=8>
<br>
Test Line<br>
<br>
<gantt
 prog="1-190,1-226"
 target="1-226,1-238,0-238,0-333,1-333,1-381,0-381,0-476,1-476,1-524"
 baseline="0-0,0-190,1-190,1-238,0-238,0-333,1-333,1-381,0-381,0-476,1-476,1-524" border=.5>
<br>
<br>
Task A10
</td></tr>
<tr>
<td>Sample section 2</td>
<td comptext=off nocomp size=8>
<br>
Test Line<br>
<br>
<gantt
 prog="1-190,1-350"
 target="1-350,1-490,0-490,0-568,1-568,1-671"
 baseline="0-0,0-190,1-190,1-350,0-490,1-490,0-490,0-568,1-568,1-671" border=.5>
<br>
<br>
Task A15
</td></tr>
</table>
```

Gantt Charts



Graphs

A graph is a raster image (jpeg) used to graphically display numeric information. See the Charts section to instead create a vector based drawing within the PDF. Graphs use the Perl GD::Graph library. The GD::Graph library is freely available for Perl. You might prefer to use this method if you have existing programming making use of using GD::Graph. Search the internet if you would like to learn more about GD::Graph. A few sample are included in the following pages to get you started.

The graph is defined using an embedded Perl script inside your input (frw) file or by using the SRC option to specify an external file. Return a jpeg of your graph as an image in memory then refer to that image in any TEXT block or table TD tag. The samples on the following pages illustrate how this done.

The -allowperl option or setAllowPerl method must be specified when creating graphs since you are executing Perl code within your report.

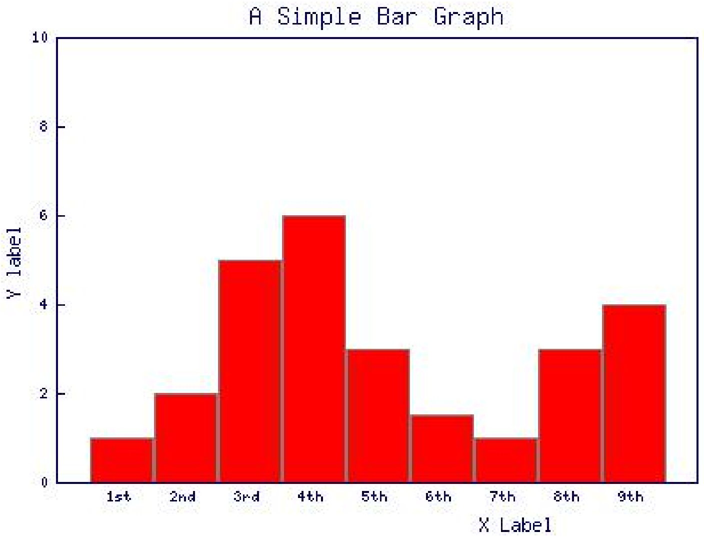
<GRAPH
NAME=text
SRC=text
ARGV=text
PERL>
</GRAPH>

The Perl code for the graph goes between the opening and closing GRAPH tags. See the example on the following page for details.

Parameter	Description
NAME=text	The name of the output you want to use to refer to the image later in the document, for example "mygraph.jpg". Treat it like you would any image file stored on disk that you might include in your PDF. You may refer to the graph as many times as you want in your document.
SRC=text	Optional. Source file containing the Perl code for the graph. Any content between the opening and closing GRAPH tags is ignored when this option is used.
ARGV=text	Optional argument to pass in to the Perl code. For example, you might have your data points in a separate file. In this case you could pass in the file name(s) to your program and access them in your code. The entry is passed as a single string so you may parse it however you want. For example, ARGV="testfile.dat testfile2.dat 'Graph Title'" would pass the string as-is. Your Perl code may then refer to $_[0] which in this case will be "testfile.dat testfile2.dat 'Graph Title'".
PERL	Required option.

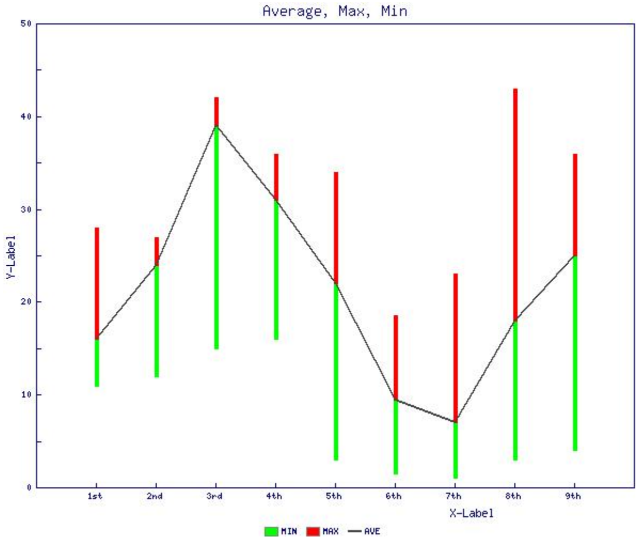
```
<PDF>
<PAGE>
<GRAPH NAME=g1.jpg PERL>
use GD::Graph::bars;
use GD::Graph::Data;

my $data = GD::Graph::Data-&lt;new([
    ["1st","2nd","3rd","4th","5th","6th","7th", "8th", "9th"],
    [    1,    2,    5,    6,    3,  1.5,    1,     3,     4],
]) or die GD::Graph::Data->error;

my $graph = GD::Graph::bars->new;

$graph->set(
    x_label         => 'X Label',
    y_label         => 'Y label',
    title           => 'A Simple Bar Graph',
) or die $graph->error;

  $graph->plot($data) or $err = @$;
  return $graph->gd->jpeg;
</GRAPH>
<TEXT>
Here is the graph: <IMG SRC="g1.jpg" HEIGHT=200 KEEPRATIO>
</TEXT>
```

Here is the graph:



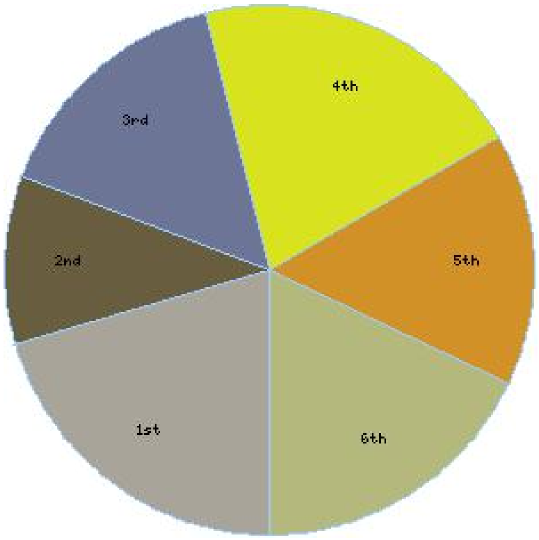
```
<PDF>
<PAGE>
<GRAPH NAME=g2.jpg PERL>
use GD::Graph::mixed;
@data = (
["1st", "2nd", "3rd", "4th", "5th", "6th", "7th", "8th", "9th"],
[11, 12, 15, 16, 3, 1.5, 1, 3, 4], # area
[5, 12, 24, 15, 19, 8, 6, 15, 21], # min block
[12, 3, 3, 5, 12, 9, 16, 25, 11],  # max block
[16, 24, 39, 31, 22, 9.5, 7, 18, 25]);
my $graph = new GD::Graph::mixed(600,500);
$graph->set(
  x_label => 'X-Label',
  y_label => 'Y-Label',
  title => 'Average, Max, Min',
  y_min_value => 0,
  y_max_value => 50,
  y_tick_number => 10,
  y_label_skip => 2,
  cumulate => 1,
  types => [qw(area bars bars lines)],
  dclrs => [undef, qw(#00ff00 #ff0000 #444444)],
  borderclrs => [undef, qw(black black black)],
  line_width => 2,
  bar_width => 4,
  transparent => 0);
$graph->set_legend(undef, qw(MIN MAX AVE));
$graph->plot(\@data);
return $graph->gd->jpeg;
</GRAPH>
<TEXT>
Here is the graph: <IMG SRC="g2.jpg" HEIGHT=200 KEEPRATIO>
</TEXT>
```

Here is the graph:



Graphs

```
<PDF>
<PAGE>
<GRAPH NAME=g3.jpg PERL>
use GD::Graph::pie;
my @data = (
            ["1st","2nd","3rd","4th","5th","6th"],
            [    4,    2,    3,    4,    3,  3.5]
        );
my $mygraph = GD::Graph::pie->new(300,300);
# colors of the pie slices
$mygraph->set( dclrs => [ "#A8A499","#685E3F","#6C7595","#D8E21F",
                          "#D19126","#B5B87D","#B7C8E2","#DFE3E1" ] );
# color of pie divisors
$mygraph->set( accentclr => '#99ccff');
$mygraph->set( '3d' =>'0');
$mygraph->plot(\@data);
return $mygraph->gd->jpeg;
</GRAPH>
<TEXT>
Here is the graph: <IMG SRC="g3.jpg" HEIGHT=200 KEEPRATIO>
</TEXT>
```

Here is the graph: 

Graphs

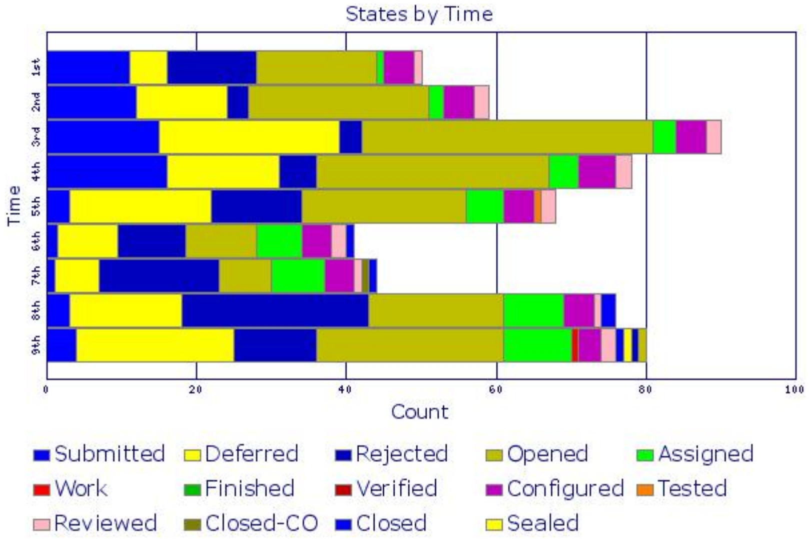
```
<PDF>
<PAGE>
<GRAPH NAME=graph.jpg PERL>
use GD::Graph;
use GD::Graph::hbars;
use GD::Image;
GD::Image->can('stringTTF') or return;
use GD::Text;
@dat = qw(
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23
);
@sub = qw(0 0 0 0 0 0 0 0 1 1 1 1 2 3 1 1 1 1 2 2 6 8 8);
@def = qw(0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0);
@rej = qw(0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0);
@opn = qw(4 4 4 5 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3);
@ass = qw(0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0);
@wrk = qw(1 2 2 2 2 2 1 1 2 2 2 1 1 1 1 1 1 1 1 1 3 6 5);
@fin = qw(0 0 0 0 0 0 1 0 0 0 0 1 1 1 2 2 2 2 2 2 2 2 2);
@ver = qw(0 0 0 0 0 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1);
@con = qw(0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0);
@tst = qw(0 0 0 0 0 0 0 0 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1);
@rev = qw(0 0 0 0 0 0 0 0 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1);
@cco = qw(0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0);
@cls = qw(0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0);
@sld = qw(0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 2 2 3 3 3 4);
# setup x data
push(@data,\@dat); # push x labels into plot data
push(@data,\@sub);
push(@data,\@def);
push(@data,\@rej);
push(@data,\@opn);
push(@data,\@ass); # push x values into plot data
push(@data,\@wrk); # (push order must match legend label order)
push(@data,\@fin);
push(@data,\@ver);
push(@data,\@con);
push(@data,\@tst);
push(@data,\@rev);
push(@data,\@cco);
push(@data,\@cls);
push(@data,\@sld);
# setup legend labels
@legend = qw(
Submitted Deferred Rejected Opened Assigned Work
Finished Verified Configured Tested Reviewed
Closed-CO Closed Sealed
);
my $graph = GD::Graph::hbars->new(600, 400);
# set graph legend
$graph->set_legend(@legend);
use GD::Text;
GD::Text->font_path('c:/windows/fonts/');
my $gd_text = GD::Text->new() or die GD::Text::error();
```

Graphs

```
$gd_text->set_font('arial.ttf', 11) or die $gd_text->error;
$graph->set_title_font( ['verdana', 'arial', gdMediumBoldFont], 11 );
$graph->set_x_label_font(['verdana', 'arial', gdMediumBoldFont], 9 );
$graph->set_y_label_font( ['verdana', 'arial', gdMediumBoldFont], 11 );
$graph->set_legend_font(
        ['verdana', 'arial', gdMediumBoldFont], 12);
# set graph options
$graph->set(
'dclrs' => [ qw(lblue lyellow blue yellow lgreen lred
green red purple orange pink dyellow) ],
'title' => "States by Time",
'x_label' => "Time",
'y_label' => "Count",
'long_ticks' => 1,
'tick_length' => 0,
'x_ticks' => 0,
'x_label_position' => .5,
'y_label_position' => .5,
'cumulate' => 1,
'bgclr' => 'white',
'transparent' => 0,
'interlaced' => 1,
'y_tick_number' => 5,
'y_number_format' => '%d',
#'y_max_value' => 25,
#'y_min_value' => 0,
'y_plot_values' => 1,
'x_plot_values' => 1,
'zero_axis' => 1,
);
$graph->set('x_labels_vertical'=> 1);
$ret = $graph->plot(\@data) or $err = @$;
return $ret->jpeg;
</GRAPH>
<TEXT>
Here is the graph: <IMG SRC="graph.jpg" HEIGHT=200 KEEPRATIO>
</TEXT>
```

Graphs

Here is the graph:



Drawing

Drawing commands are used to draw lines, rectangles and circles. These commands are typically placed outside of TEXT or TABLE structures.

You may use LINE, RECT, CIRCLE and ARC in table cells however. See the notes on these commands as there are considerations to how they are used in this case. Cells are not sized based on the height or width of these drawing commands so you may need to use the WIDTH option on the cell and/or set a font size large enough for the height even if no text is drawn. For example, you might place <FONT SIZE=30> as the first command inside a table cell in order to force the height to be at least 30 points.

Drawing

```
<ARC
    X=number
    Y=number
    RADIUS=number|numberI
    DEG1=number
    DEG2=number
    ON=number
    OFF=number
    DASH=number,number[,number,number...]
    WIDTH=number
    FCOLOR=color
    SCOLOR=color
    FIXED>
```

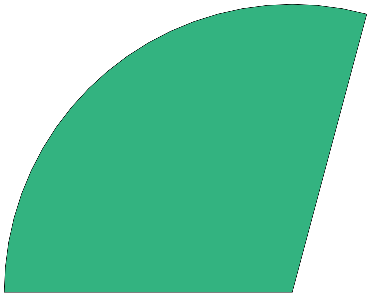
Used to draw an arc.

Parameter	Description
X=number	X point based on the GRID command. For tables, X is a percentage of the cell width when used inside of a TD tag. To force the X position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.
Y=number	Y point based on the GRID command. For tables, this value is added to the top of the current cell when used inside of a TD tag.
RADIUS=number|numberI	Radius based on the GRID command. If you put a letter I after the number then radius is expressed in inches.
DEG1=number	Starting degree value (from 0 to 360).
DEG2=number	Ending degree value (from 0 to 360).
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to leave blank.

Drawing

Parameter	Description
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
FIXED	For arcs drawn in table cells. To force the X position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.

For example, the following arc is drawn from 0 to 105 degrees.



Drawing

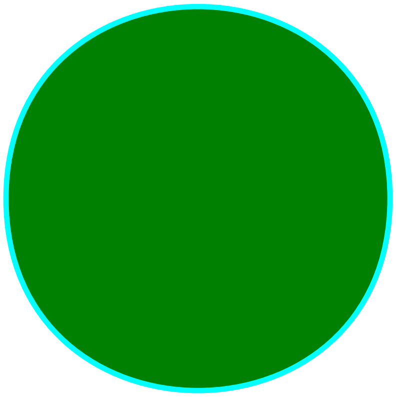
<CIRCLE
X=number
Y=number
ON=number
OFF=number
DASH=number,number[,number,number...]
RADIUS=number|numberI
WIDTH=number
FILL
FCOLOR=color
SCOLOR=color
SHADING=text
FIXED>

Used to draw a circle.

Parameter	Description
X=number	X center point based on the GRID command. For tables, X is a percentage of the cell width when used inside of a TD tag. To force the X position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.
Y=number	Y center point based on the GRID command. For tables, this value is added to the top of the current cell when used inside of a TD tag.
RADIUS=number|numberI	Radius based on the GRID command. If you put a letter I after the number then radius is expressed in inches (centimeters when METRIC is used).
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to leave blank.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.

Parameter	Description
FILL	If set fills the circle with the fill color.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
SHADING=text	A shading pattern to use for the background. See an example here.
FIXED	For circles drawn in table cells. To force the X position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.

For example, the following circle is created using <CIRCLE X=40 Y=20 RADIUS=1I SCOLOR=aqua FCOLOR=green WIDTH=2 FILL>.



Drawing

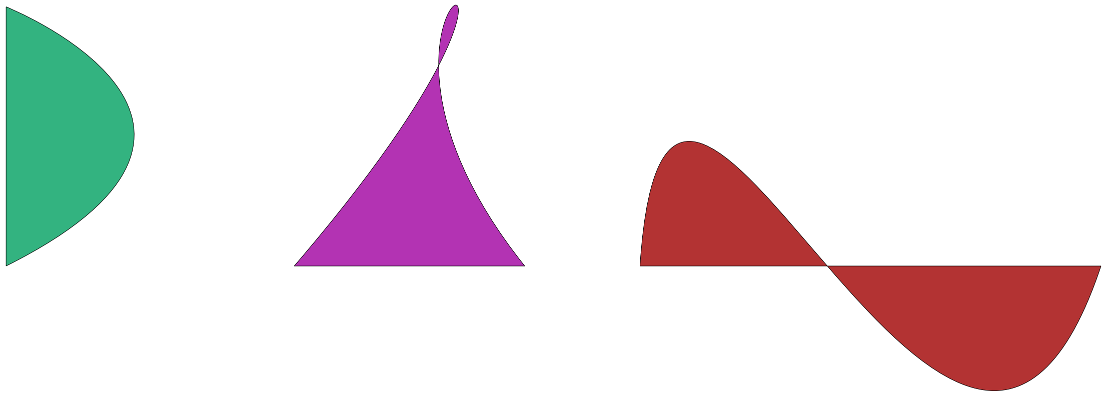
<CURVE
X1=number
Y1=number
X2=number
Y2=number
CPX1=number
CPY1=number
CPX2=number
CPY2=number
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
FILL
FCOLOR=color
SCOLOR=color>

Used to draw a closed curve.

Parameter	Description
X1=number	X GRID position of first point.
Y1=number	Y grid position of first point.
X2=number	X grid position of second point.
Y2=number	Y grid position of second point.
CPX1=number	X grid position of first control point.
CPY1=number	Y grid position of first control point.
CPX2=number	Optional. X grid position of second control point.
CPY2=number	Optional. Y grid position of second control point.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to leave blank.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.

Parameter	Description
FILL	If set fills the curve with the fill color.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.

Here are some examples:



<LINE
X1=number
Y1=number
X2=number
Y2=number
Y=number
BLANK
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
LINEC=number
FCOLOR=color
SCOLOR=color
FIXED>

Used to draw a line. You can also use HR in place of LINE. This allows you to draw a horizontal rule like you would in HTML.

Parameter	Description
X1=number	From X point based on the GRID command. For tables, X1 is a percentage of the cell width when used inside of a TD tag. To force the X1/X2 position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.
Y1=number	From Y point based on the GRID command. For tables, this value is added to the top of the current cell when used inside of a TD tag.
X2=number	Thru X point based on the GRID command. For tables, X2 is a percentage of the cell width when used inside of a TD tag. To force the X1/X2 position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.
Y2=number	Thru Y point based on the GRID command. For tables, this value is added to the top of the current cell when used inside of a TD tag.

Drawing

Parameter	Description
Y=number	The Y value in points from the bottom of the cell. This is only used with the BLANK option.
BLANK	This gives the line special meaning when used inside of a table cell. Use it when you want to draw a line from the end of the text to the right side of the table cell. A form that someone will print to fill out various bits of information such as date, name, age, etc. would be one example. In this case you could specify your text along with a line to write the information on. The structure would be similar to this: <TD>First Name<LINE X1=1 X2=0 Y=-1 BLANK></TD>. You may need to specify a width on the TABLE or TD tag to leave enough room for a long enough line. The X1 value for the LINE tag in this case means the number of points (1/72 of an inch) to leave after the text before the line is started. The X2 value is in points from the right edge of the cell to draw the line. The Y value is in points from the bottom of the cell.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
LINEC=number	Sets the line end style 0 = Butt ends 1 = Round ends 2 = Projecting square caps

Parameter	Description
FCOLOR=color	Sets the fill color. Only used when drawing dashed lines.
SCOLOR=color	Sets the stroke color. This is the color used for the line.
FIXED	For lines drawn in table cells. To force the X1/X2 position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.

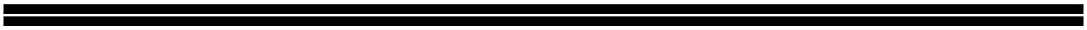
You may leave the Y1 and Y2 positions out to simply draw a horizontal rule at your current position. For example, the following line is positioned just below this text by using <LINE X1=10 X2=70 SCOLOR=green WIDTH=2>.

<LINEC
VALUE=number>

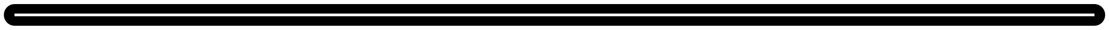
Used to set the line caps.

Parameter	Description
VALUE=number	Sets the line end style 0 = Butt ends 1 = Round ends 2 = Projecting square caps

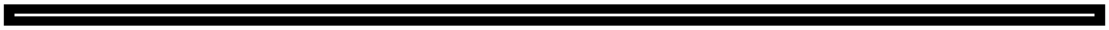
Here a an example with 0



Here a an example with 1



Here a an example with 2



<LINED
ON=number
OFF=number
DASH=number,number[,number,number,...]>

Used to set the on/off pixels for line drawing. Note that the "on" color is set with the SCOLOR value. The line won't appear dashed if both FCOLOR and SCOLOR are the same.

Parameter	Description
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to draw with the FCOLOR.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length. The DASH option and the ON/OFF options are mutually exclusive.

The line below is drawn with ON=8, OFF=4, FCOLOR=white and SCOLOR=green.

**<LINEW
VALUE=number>**

Sets the line thickness. Default is 1.

Parameter	Description
VALUE=number	The width of the line.

Here a width of .1

Here a width of 1

Here a width of 5

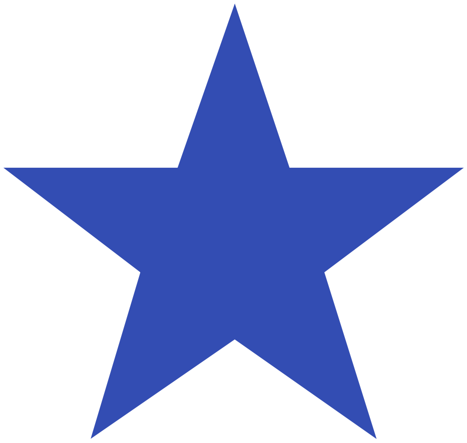
Drawing

<POLY
POINTS=number,number[,number]
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
FILL
FCOLOR=color
SCOLOR=color>

Used to draw a polygon.

Parameter	Description
POINTS=number, number [,number]	Series of points in the form X1,Y1,X2,Y2,... as based on the current GRID settings.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to leave blank.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FILL	If set fills the polygon with the fill color.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.

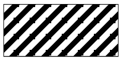
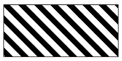
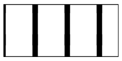
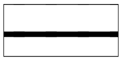
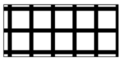
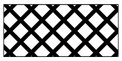
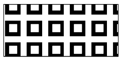
For example, the following polygon is created using <POLY POINTS=100,2,77,68,7,68,62,110,42,177,100,137,157,177,136,110,192,68,122,68 WIDTH=2 FCOLOR=.2,.3,.7 SCOLOR=.2,.3,.7 FILL>.



<RECT
X1=number
Y1=number
X2=number
Y2=number
ON=number
OFF=number
DASH=number,number[,number,number...]
WIDTH=number
LINEC=number
FILL
FCOLOR=color
SCOLOR=color
SHADING=text
PATTERN=number
CORNERSIZE=number
CORNERSTYLE=text
FIXED>

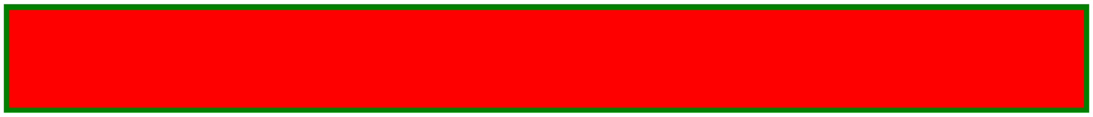
Used to draw a rectangle.

Parameter	Description
X1=number	From X point based on the GRID command. For tables, X1 is a percentage of the cell width when used inside of a TD tag. To force the X1/X2 position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.
Y1=number	From Y point based on the GRID command. For tables, this value is added to the top of the current cell when used inside of a TD tag.
X2=number	Thru X point based on the GRID command. For tables, X2 is a percentage of the cell width when used inside of a TD tag. To force the X1/X2 position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.

Parameter	Description
Y2=number	Thru Y point based on the GRID command. For tables, this value is added to the top of the current cell when used inside of a TD tag.
ON=number	The length of the line segment to draw with the SCOLOR.
OFF=number	The length of the line segment to leave blank.
DASH=number,number[,number,number,...]	A comma separated list of numbers (in pairs) to specify the ON/OFF length.
WIDTH=number	Sets the line width based on units of 1/72 of an inch.
FILL	If set fills the rectangle with the fill color.
FCOLOR=color	Sets the fill color.
SCOLOR=color	Sets the stroke color.
SHADING=text	A shading pattern to use for the background. Set to the NAME value from the SHADING tag.
PATTERN=number	A patten for the interior. Set to a value from 1 to 8. 1 =  2 =  3 =  4 =  5 =  6 =  7 = 

Parameter	Description
	8 = ▦
CORNERSIZE=number	The size of the optional CORNERSTYLE in inches (centimeters if METRIC is used).
CORNERSTYLE=text	The optional type of corners to use for the rectangle. Valid values are: Round Bevel Scoop Stair In addition, you may specify a different style for any given corner. Pass a comma separated string containing a code for each corner followed by a colon then the type. The codes for each corner are: TL - Top left TR - Top right BR - Bottom right BL - Bottom left For example, set to "TL:round,TR:scoop,BR:round" to give the top left and bottom right corners a round edge and the top right a scoop edge. The bottom left edge, since it wasn't specified, will have a standard right-angled corner.
FIXED	For rectangles drawn in table cells. To force the X1/X2 position based on the current GRID inside of TD cells (rather than based on current cell X positioning) include FIXED.

You may leave the Y1 position out to draw a rectangle at the current position. The value for Y2 will then be added to the current position rather than used as an absolute position. For example, the following rectangle is positioned just below this text by using <RECT X1=10 X2=70 Y2=5 SCOLOR=green FCOLOR=red WIDTH=2 FILL>.



Interactive Forms

There are a variety of widgets you can place into your PDF for creating interactive forms. Users of the full version of Acrobat (not the free Reader) will be able to save data they enter. Users of Adobe Reader will be able to enter and print the information but will not be able to save the PDF with their data. The following section describes the various widgets available.

The widgets are placed within TEXT tags or within any text printed in a table TD cell. Be sure to NAME EACH FIELD with a unique name. If you omit the NAME parameter or use duplicate names without the EXT option, you'll generate invalid PDFs.

The EXT option allows you to duplicate a field in more than area of the PDF. You may want to have a prompt for "name" that appears on two different pages of the PDF and not have the user enter his or her name in both fields. The EXT option denotes the "array extent" of the field. The fields must be of the same type (like TEXT or CHECKBOX) but do not need to have the same appearance in terms of font or point size. Simply name the fields the same and assign a unique value (within that group) to EXT for each field.

```
<INPUT
    TYPE=text
    SUBMITTYPE=text
    NAME=text
    EXT=number
    DESCR=text
    SIZE=number
    MAXLENGTH=number
    ALIGN=L|C|R
    WIDTH=number
    HEIGHT=number
    ROTATE=number
    LOCK[=list]
    LOCKEXCEPT=list
    FCOLOR=color
    BGCOLOR=color
    BORDERCOLOR=color
    FACE=number
    POINTSIZE=number
    RAISED=number|SUNKEN=number
    NUMBER
    NOCOMMAS
    DEC=number
    DATE[=text]
    CHECKED
    SYMBOL
    RANGE=name
    READONLY
    REQUIRED
    ONCLICK=text
    ONCHANGE=text
    EXCLUDE
    URL=text
    IMG=text
    VALUE=text
    SIGNSSL="text"
    SIGNPKFILE="text"
    SIGNPEMFILE="text"
    SIGNTIMESTAMP="text"
    SIGNTS="text"
    SIGNREASON="text"
    SIGNIMG="text"
    SIGNKEEPRATIO
    SIGNIMGALIGN="text"
    SIGNSIZE=number
    SIGNBGCOLOR="text"
    SIGNNOSTATUS>
```

Used to insert a text input line or checkbox widget.

Parameter	Description
TYPE=TEXT | HIDDEN | SIGNATURE | CHECKBOX | RADIO | BUTTON | SUBMIT	Set to TEXT for a text entry box, HIDDEN for a hidden field, SIGNATURE for a signature, CHECKBOX for a checkbox, RADIO for radio button, BUTTON for a button or SUBMIT for a Submit button. Hidden fields are mainly used to pass data back to a web site without the data actually displayed in the PDF (like a customer number, for example). Note that users will need the full version of Acrobat (not the free Reader) or FyTek's PDF Meld in order to electronically sign documents. Signing involves changes to the PDF which is not a function of Adobe Reader.
SUBMITTYPE=XFDF | FDF (default) | PDF | GET | POST	The type of contents to send for a submit button. Use this option when TYPE=SUBMIT. Set to XFDF, FDF or PDF to attach the corresponding document type to an email if using mailto: as the URL. Set to GET or POST if sending the data to a website.
NAME=text	Assign each input statement a unique name or have a different EXT for the field. This can be a number or text - do not use spaces. For a radio button group, use the same name for each button in the group but use a different VALUE for each one.

Parameter	Description
EXT=number	Allows setting up the same named field in multiple areas of the PDF. Only valid for TEXT, CHECKBOX and TEXTAREA widgets. Assign the same name to two or more text fields while using a different EXT value for each. The value input into one field will be carried over to all with the same name.
DESCR=text	Optional description to be used when generating error or status messages about the field.
SIZE=number	The width in characters of the text entry line (this will be approximate).
MAXLENGTH=number	The maximum number of characters allowed in a text entry line (optional).
ALIGN=L|C|R	Sets the alignment of text entry to Left, Right or Center. Default is Left.
WIDTH=number	Sets the width based on grid units (rather than the SIZE value).
HEIGHT=number	Sets the height in lines based on current point size. This is mainly for the signature type field. The signature field usually holds several lines of text when filled in so it's best to set this to a value somewhere between 3 and 5.
ROTATE=number	Use this option to rotate the form field. This is useful in cases where your background PDF contains a rotation setting and you want the field orientation to match. Valid values are 0 (the default), 90, 180, or 270.

Parameter	Description
LOCK[=list]	Used to lock down fields in the PDF after signing. Use just LOCK by itself to lock down all fields or supply a list of field names, separated with a comma, that you want locked down.
LOCKEXCEPTION=list	Used to lock down fields in the PDF after signing except the ones listed. This option is mutually exclusive from the LOCK command above. The LOCK command takes precidence when both LOCK and LOCKEXCEPTION are used.
FCOLOR=color	The text color for the widget.
BGCOLOR=color	The background color for the widget.
BORDERCOLOR=color	The border color for the widget.
FACE=number	The font number to use (2 is the default) for a text, textarea, button or select widget. The valid values are: 1 = Courier 2 = Helvetica 3 = Times Roman
POINTSIZE=number	The font size to use for a text, textarea, button or select widget. You may use a value of 0 for text or textarea to signify autosize. Autosize will automatically shrink text to fit within the bounding box of the text widget.

RAISED=number|
SUNKEN=number
	Used to display either a raised or sunken border around a text or checkbox widget. The number is the width of the border in units of 1/72 of an inch. A typical value would be 2. Keep in mind the user can set any magnification level for the PDF and the effect may not look good at all levels.

Parameter	Description
NUMBER	Allows only numbers to be entered in the text widget.
NOCOMMAS	Turns off the display of commas in a text entry widget marked as a NUMBER.
DEC=number	Sets the number of decimal places in a text entry widget marked as a NUMBER.
DATE[=text]	Sets the text entry widget as a date/time field. The default format is mm/dd/yyyy. You can optionally set DATE to a string that represents the date format you want. Some examples are: yyyy/mm/dd yyyy/m/d d-m-yyyy mmm-d-yyyy mmmm d, yyyy mmmm d, yyyy hh:MM tt HH:MM
CHECKED	Defaults the checkbox or radio button to a checked state.
SYMBOL	The ASCII character to use for the check in the checkbox from the Zapf Dingbats character set (default is 4).
RANGE=name	Sets the name of the page range to use for a button. See the FROMPG and THRUPG commands for setting up a page range. The button will print the page range to the user's default printer when clicked. There is no user intervention for printer selection or other options. This option overrides the ONCLICK input option. The value is case sensitive.
READONLY	Sets the widget to read only status.

Parameter	Description
REQUIRED	Forces user to enter a value before submitting.
ONCLICK=text	Set the JavaScript for the onClick event for the button.
ONCHANGE=text	Set the JavaScript for the onChange event for the text widget. The events in PDF are slightly different from HTML so technically this is the onBlur event for the PDF field (triggered when the item loses focus).
EXCLUDE	Excludes the input/value pair when submitting to a web site. The default is to submit all input/value data.
URL=text	Sets the URL the form should submit its data to (for example, http://www.site.com/prog.cgi or mailto:user@somesite.com). Use the SUBMITTYPE to define what contents you want to pass along. For an HTTP site you'll most likey want to use GET or POST. For email, use FDF, XFDF or PDF.
IMG=text	Use the specified image in place of text for a button widget. Button widgets will display the VALUE text when this option is left off.
VALUE=text	Initial text to display in a text entry widget.
SIGNSSL="text"	Used when signing a signature field. The path and file name of the OpenSSL program. For example, "c:\openssl\bin\openssl.exe". See the Digital Signature section for details.

Parameter	Description
SIGNPKFILE="text"	Used when signing a signature field. The path and name of the private key file. For example, "c:\keys\mykey_pk.pem". See the Digital Signature section for details.
SIGNPEMFILE="text"	Used when signing a signature field. For example, "c:\keys\mykey.pem". The path and name of the der-encoded signing certificate. See the Digital Signature section for details.
SIGNTIMESTAMP="text"	The path and name of your timestamp server if you want to include an embedded timestamp on the signature. Use "1" to use the defaul of https://freetsa.org/tsr. Pass your user name and password if your timestamp server requires it like this: "myuserid:mypassword@https://mytimestamp.org". Use the SIGNTS option with SIGNTIMESTAMP.
SIGNTS="text"	The path and name of the ts.exe (or just ts for Linux) executable that is used to perform the timestamping function. This is a separate download available here. Use this program for both Windows and Linux.
SIGNREASON="text"	Used when signing a signature field. Optional. The reason for signing the document. Default is "Attestation to the accuracy and integrity of this document". See the Digital Signature section for details.

John Doe

123 Main Street

Parameter	Description
SIGNIMG="text"	Used when signing a signature field. Optional. The path and [illegible] image to use for the signature. Set this option to the value "none" to not place any image in the signature field. See the Digital Signature section for details.
SIGNKEEPRATIO	Used when signing a signature field. Optional. Keep the image x/y scaling ratio when using an image with a signature field. See the Digital Signature section for details.
SIGNIMGALIGN="text"	Optional. The alignment for the image in the signature field when using SIGNKEEPRATIO. Set to "Left", "Center", or "Right". Default is "Left". See the Digital Signature section for details.
SIGNSIZE=number	Used when signing a signature field. Optional. The point size for the font in the text of the signature (or 0 for no text). Default is 12. See the Digital Signature section for details.
SIGNBGCOLOR="text"	Used when signing a signature field. Optional. The background fill color for the signature. See the Digital Signature section for details.

Here are some examples:

Type	Sample
Text entry	Name Address City State Zip Dup. Name John Doe
Signature	Sign here You can use FyTek's PDF Meld or the full version of Acrobat in order to electronically sign documents.
Checkbox	□ Option A ☑ Option B 🟨 Option C
Radio Set	● Option A ○ Option B ○ Option C
Button	Value 1 Value 2 Result Add

<SCRIPT>
</SCRIPT>

Used to define global JavaScript functions for the PDF. You can define a function here then reference it from a widget. For information on the JavaScript language, see Adobe's JavaScript Object Specification (#5186).

Place your functions within the SCRIPT opening and closing tags like this:

```
<SCRIPT>
function funct1(parm1, parm2)
{
  var parm3 = parm1 + parm2;
  var abcFld = this.getField("abc");
  abcFld.value = parm3;
}
function getMax(x, y, z)
{
  var maxval = (x > y ? x : y);
  maxval = (z > maxval ? z : maxval);
  return maxval;
}
</SCRIPT>
```

Then reference these fuctions from any widget in the ONCHANGE or ONCLICK parameter of the tag. Note that you may break up your functions across lines but, since a tag cannot be broken up, the ONCHANGE or ONCLICK parameters must be written out as one line. Using the SCRIPT tags and referencing just a function will keep the widget tags smaller and more readable rather than trying to insert lots of commands in the ONCHANGE or ONCLICK paramters.

```
<SELECT
    NAME=text
    DESCR=text
    ENTRY
    WIDTH=number
    ROTATE=number
    FCOLOR=color
    BGCOLOR=color
    BORDERCOLOR=color
    ONCHANGE=text
    EXCLUDE
    READONLY
    REQUIRED>

<OPTION VALUE=text_1>Display Text 1
<OPTION VALUE=text_2>Display Text 2
. . .
<OPTION VALUE=text_n>Display Text n

</SELECT>
```

Used to insert a selection widget. Place individual list items in the OPTION tags.

Parameter	Description
NAME=text	Assign each select a unique name. This can be a number or text - do not use spaces.
DESCR=text	Optional description to be used when generating error or status messages about the field.
ENTRY	Allows the user to type in a value not in the list.
WIDTH=number	Sets the width based on grid units (rather than the SIZE value).
ROTATE=number	Use this option to rotate the form field. This is useful in cases where your background PDF contains a rotation setting and you want the field orientation to match. Valid values are 0 (the default), 90, 180, or 270.
FCOLOR=color	The text color for the widget.
BGCOLOR=color	The background color for the widget.
BORDERCOLOR=color	The border color for the widget.

Parameter	Description
ONCHANGE=text	Set the JavaScript for the onChange event. The events in PDF are slightly different from HTML so technically, this is the onBlur event for the PDF field (triggered when the item loses focus).
EXCLUDE	Excludes the input/value pair when submitting to a web site. The default is to submit all input/value data.
READONLY	Sets the widget to read only status.
REQUIRED	Forces user to enter a value before submitting.

Here are some examples:

State	California
Choose one or key in your own text	Transfer to Checking

```
<TEXTAREA
    NAME=text
    EXT=number
    DESCR=text
    ROWS=number
    COLS=number
    MAXLENGTH=number
    WIDTH=number
    ROTATE=number
    FCOLOR=color
    BGCOLOR=color
    BORDERCOLOR=color
    ONCHANGE=text
    EXCLUDE
    READONLY
    REQUIRED>
</TEXTAREA>
```

Used to insert a multi-line text input widget. Any default value goes in-between the opening and closing tag.

Parameter	Description
NAME=text	Assign each textarea a unique name. This can be a number or text - do not use spaces.
EXT=number	Allows setting up the same named field in multiple areas of the PDF. Assign the same name to two or more text area fields while using a different EXT value for each. The value input into one field will be carried over to all with the same name.
DESCR=text	Optional description to be used when generating error or status messages about the field.
ROWS=number	The height in characters of the entry box (this will be approximate).
COLS=number	The width in characters of the entry box (this will be approximate).
MAXLENGTH=number	The maximum number of characters allowed in a text entry box (optional).
WIDTH=number	Sets the width based on grid units (rather than the SIZE value).

Parameter	Description
ROTATE=number	Use this option to rotate the form field. This is useful in cases where your background PDF contains a rotation setting and you want the field orientation to match. Valid values are 0 (the default), 90, 180, or 270.
FCOLOR=color	The text color for the widget.
BGCOLOR=color	The background color for the widget.
BORDERCOLOR=color	The border color for the widget.
ONCHANGE=text	Set the JavaScript for the onChange event for the text or text area widget. See Adobe's JavaScript Object Specification (#5186). The events in PDF are slightly different from HTML so technically, this is the onBlur event for the PDF field (triggered when the item loses focus).
EXCLUDE	Excludes the input/value pair when submitting to a web site. The default is to submit all input/value data.
READONLY	Sets the widget to read only status.
REQUIRED	Forces user to enter a value before submitting.

Here are some examples:

Comments	
Additional Information	Here is some default text for this entry box.

Digital Signatures

Reasons for Using

Digital signatures provide a way to sign a PDF electronically in order to authenticate its contents. Signing a PDF electronically is a process that creates a unique encoding out of the entire PDF. No two PDFs will have the same encoding unless they are exactly the same. This encoding is then signed (further encoded) by running a program that takes as input the encoding, a public key signature file, and a private key password to create the PDF signature. The end user can verify the PDF is authentic by checking that the applied signature is valid. Adobe Reader or Acrobat will recompute the encoding over the entire PDF and if the signature in the PDF does not match then the signature is no longer valid.

Most casual users of PDF familiar with Adobe Reader mistakenly believe that PDFs cannot be modified. In reality, it's fairly easy to change text in a PDF using Adobe Acrobat or other third-party programs like FyTek's PDF Meld. A digital signature is therefore used not to prevent changes but to validate that a PDF you are viewing has not been altered or, if altered and the signature is still valid, what changes were made. Any changes to the document after the signature is applied will be noted in the signature pane in Reader or Acrobat so you know what was modified. Allowed changes could be a situation where you have two signature fields and you have one signed and the other is blank. The act of signing the second signature field alters the document but can do so in such a way as to keep the initial signature valid. Also, no unauthorized user can use your signature information from a signed PDF to sign with themselves as they will not have your private signing key file nor your private password.

For example, you may have a contract that both you and your client will sign. You create a PDF with two signature fields and sign one of them directly from PDF Report Writer at the time the PDF is created by passing in the signing information on the INPUT tag. Note that PDF Report Writer allows only one signature field to be signed when creating the PDF. The client can then use Adobe Acrobat or similar software that is designed for signing PDFs to apply their signature. FyTek does offer another product, PDF Meld, that can be used to sign additional signature fields at a later point. Also, PDF Meld is able to fill in fields at the time of signing if desired or add signature fields while keeping all existing signatures intact and valid.

Requirements for Signing

PDF Report Writer uses an open source program called OpenSSL that is available for most Unix and Windows installations. If you do not have OpenSSL installed, you'll need to install it first before you can digitally sign documents. Most Unix systems will likely have it - if you're not sure, try typing OpenSSL at a shell prompt and if it comes back with a prompt that looks like OpenSSL> then it is installed. Windows binaries are available here: https://www.slproweb.com/products/Win32OpenSSL.html or, if the link is not available, search for "openssl windows binary" in your favorite search engine.

The following section deals with using OpenSSL to create your signature files. While you don't need to be an expert at digital certificates, you should be comfortable running commands from the DOS prompt. This is a process you will probably only run once to set your certificates. Once you have them you simply supply them to the PDF Report Writer for signing so this section is not something you will need to do each time you want to sign a PDF.

First you'll need a certificate to sign PDFs with. You may purchase them from security companies on-line or use OpenSSL to create your own. There are 2 files you'll need to sign with:
mykey_pk.pem - your private signing key
mykey.pem - your certificate and public key

There are two types of file formats for certificate files. One is PEM which is a text file and DER which is binary. The names of your files may be different but the point is you'll need a private key in pem format and a certificate in binary form.

Covering all the commands of OpenSSL is beyond the scope of this document. We'll just be covering the basics to get a certificate setup. There are many websites to explain other uses and options for OpenSSL if you are interested. Installing OpenSSL should only take a few minutes depending on your internet connection.

Setting up OpenSSL

At this point you should have OpenSSL installed - use the link mentioned in the previous section if you need to install on Windows. The first step is to create a configuration file for OpenSSL if you don't have one already. You only need to do this once and you may place it in the directory you installed OpenSSL into. Here's a sample openssl.cnf file to get you started if you need one. This file is also embedded in this PDF so you can download it from this document rather than cut & paste.

```
#
# SSLeay example configuration file.
# This is mostly being used for generation of certificate requests.
#
RANDFILE                = .rnd
####################################################################
[ ca ]
default_ca      = CA_default            # The default ca section
####################################################################
```

```
[ CA_default ]
dir             = demoCA                # Where everything is kept
certs           = $dir\certs            # Where the issued certs are kept
crl_dir         = $dir\crl              # Where the issued crl are kept
database        = $dir\index.txt        # database index file.
new_certs_dir   = $dir\newcerts         # default place for new certs.
certificate     = $dir\cacert.pem          # The CA certificate
serial          = $dir\serial              # The current serial number
crl             = $dir\crl.pem             # The current CRL
private_key     = $dir\private\cakey.pem   # The private key
RANDFILE        = $dir\private\private.rnd # private random number file
x509_extensions = x509v3_extensions     # The extentions to add to the cert
default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # which md to use.
preserve        = no                    # keep passed DN ordering
# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_match
# For the CA policy
[ policy_match ]
countryName             = optional
stateOrProvinceName     = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_min                 = 2
countryName_max                 = 2
stateOrProvinceName             = State or Province Name (full name)
localityName                    = Locality Name (eg, city)
0.organizationName              = Organization Name (eg, company)
organizationalUnitName          = Organizational Unit Name (eg, section)
commonName                      = Common Name (eg, your website's domain name)
commonName_max                  = 64
emailAddress                    = Email Address
```

```
emailAddress_max                = 40
[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20
[ x509v3_extensions ]
# under ASN.1, the 0 bit would be encoded as 80
nsCertType                      = 0x40
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName
#nsCertSequence
#nsCertExt
#nsDataType
```

The end user will not need to do anything special to use certificates you create but they will not be trusted certificates. They have the option to trust your certificate, if they wish, but they do not have to. In either case, Reader will report whether or not the document has been modified since it was signed.

Creating Self-Signed Certificates

Now that OpenSSL is setup, here are the steps to create a self-signed certificate. Note that there are a variety of security companies that sell self-signed certificates. However, we'll use OpenSSL here to show you how to create your own in just a few short steps.

1. Open a DOS window or a shell in Linux/Unix.

2. Be sure your PATH environment variable contains the executable for OpenSSL. This will be the directory you installed it into. If not set, you can type this at the DOS prompt:
   `path=%path%;c:\(openssl-directory)`
   Where the "(openssl-directory)" is replaced with the directory containing the binary openssl.exe program. This should be the directory you installed the program into along with the path of \bin at the end of that.

3. Create the public and private key files by running the following:
   `openssl req -x509 -new -config openssl.cnf -days 365 -out mykey.pem -keyout mykey_enc.pem -newkey rsa:2048`
   The file mykey_end.pem is the encrypted private key. You may set the number of days for expiration to whatever you want. In the example, we've used 1 year but you may set for whatever you like. This is just the expiration for the certificate. Be sure to put the full path to openssl.cnf in the line above if it is not in your current directory. The -newkey rsa:2048 (or rsa:4096) is optional if you want to create larger encryption keys than the 1024 default size. Now you should have mykey_enc.pem and mykey.pem on your system.

4. Convert the encrytped private key to an RSA private key file:
   `openssl rsa -in mykey_enc.pem -out mykey_pk.pem`
   The file mykey_pk.pem is the private key you'll use for the SIGNPKFILE option. Be sure to put the full path to openssl.cnf in the line above if it is not in your current directory.

Passing Signature Information

Now that you have the files, use them on the command line or the INPUT tag of your signature field in your input file to PDF Report Writer. On the command line, -signpkfile would be set to mykey_pk.pem and -signpemfile would be mykey.pem using the example above. You do not have to pass the signing password to the program.

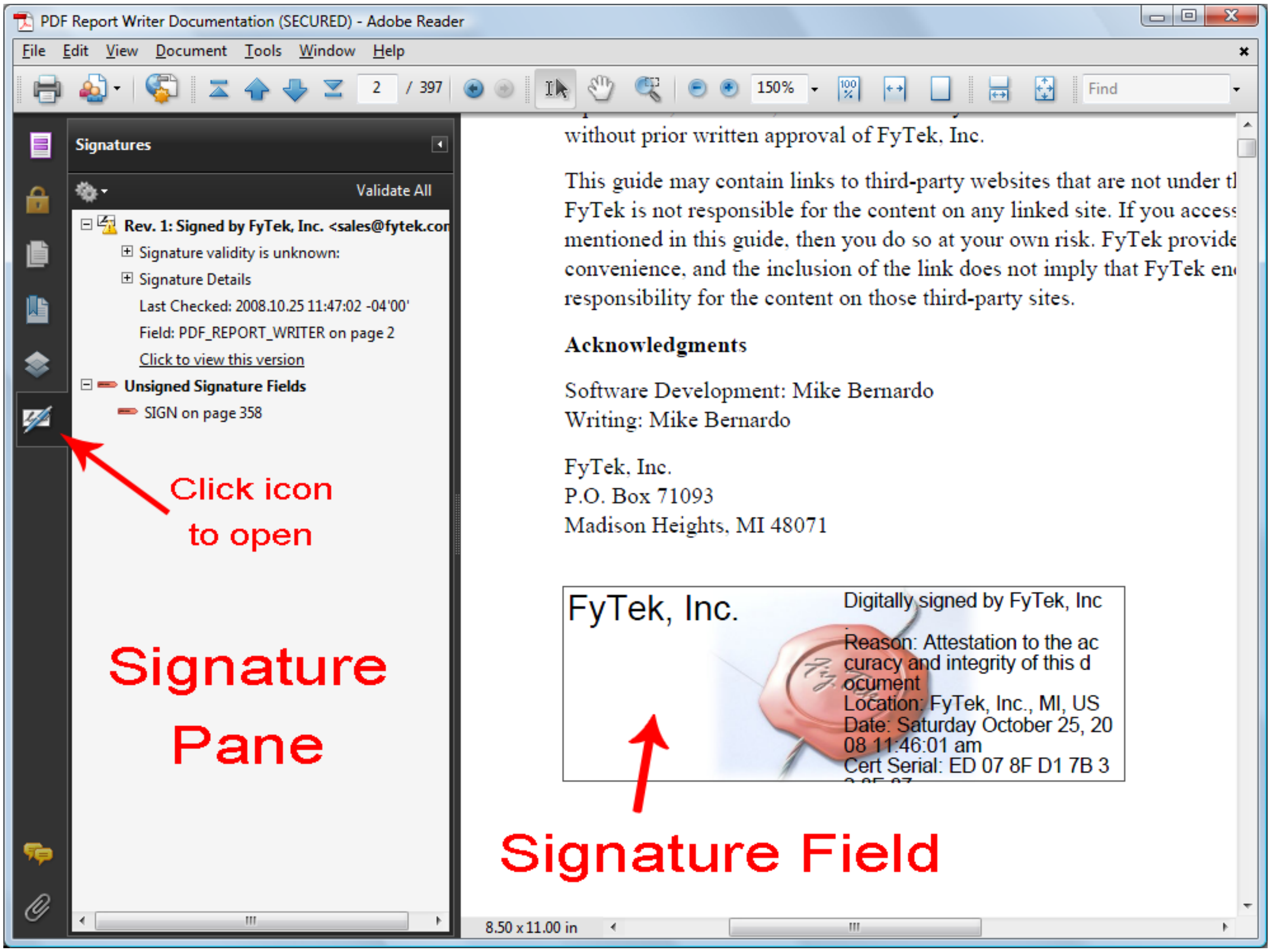
Here is a sample input signature field. This will add a signature field to a PDF and sign the field at the same time:

```
<INPUT TYPE="signature" NAME="sig"
 HEIGHT=10 WIDTH=20
 SIGNSSL="c:\openssl\bin\openssl.exe"
 SIGNPKFILE="c:\openssl\mykey_pk.pem"
 SIGNPEMFILE="c:\openssl\mykey.pem"
 SIGNREASON="Approving this document"
 SIGNIMG="c:\images\sigimage.jpg">
```

Trusting Certificates

You will see something similar to the following after you sign a document for the first time.

Digital Signatures


PDF Report Writer Documentation (SECURED) - Adobe Reader
File Edit View Document Tools Window Help
2 / 397
150%
Find
Signatures
Validate All
8.50 x 11.00 in

Rev. 1: Signed by FyTek, Inc. <sales@fytek.con
Signature validity is unknown:
Signature Details
Last Checked: 2008.10.25 11:47:02 -04'00'
Field: PDF_REPORT_WRITER on page 2
Click to view this version
Unsigned Signature Fields
SIGN on page 358

without prior written approval of FyTek, Inc.

This guide may contain links to third-party websites that are not under tl
FyTek is not responsible for the content on any linked site. If you access
mentioned in this guide, then you do so at your own risk. FyTek provide
convenience, and the inclusion of the link does not imply that FyTek en
responsibility for the content on those third-party sites.

Acknowledgments

Software Development: Mike Bernardo
Writing: Mike Bernardo

FyTek, Inc.
P.O. Box 71093
Madison Heights, MI 48071

FyTek, Inc.

Digitally signed by FyTek, Inc
Reason: Attestation to the ac curacy and integrity of this d ocument
Location: FyTek, Inc., MI, US
Date: Saturday October 25, 20 08 11:46:01 am
Cert Serial: ED 07 8F D1 7B 3

Click icon to open

Signature Pane

Signature Field

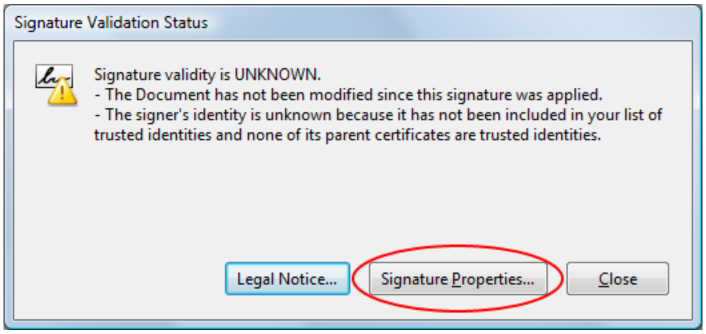
Note the icon with the yellow warning icon in the signature pane. This is because the certificate has not yet been trusted by Reader. Once you have trusted the certificate the icon will change and all future signatures in PDFs with this certificate will be recognized. There are two ways to do this.

The first method is to export the certifcate as an FDF file and send it to the intended user. You are sending just a certifcate for the user to load into Acrobat or Reader in this case, not a PDF. Also, you only need to export a given certificate one time so once you create the file you can supply the same one to end users as often as you like and it will be good not just for this PDF but for any that you sign with this certificate. To do this, open the PDF with the newly added signature. Click on the signature to open a dialog box then click the "Signature Properties..." button to continue. Next, click the "Show Certificate..." button to continue. Screen shots of these dialog windows are shown in the next section. From the certificate screen you will see a button labeled "Export...". Click the "Export..." button to step through the process of exporting the certificate.

By default you should have Acrobat FDF Data Exchange as the format for the certificate, which is what you want. Click the "Next" button and supply the requested data when prompted to create the file. Once you are finished you should have a file with an extension of .fdf you can supply to end users containing your certificate. They open this file using Acrobat or Reader and, rather than opening a PDF, are presented with a dialog box where they can step through the process of trusting the certificate. In this way you can email, place on a website, or otherwise provide the certificate in advance rather than have the user trust the certificate directly from the PDF which involves a warning dialog when attempted.

The second method is to trust the certificate directly from the PDF. You can use this method to trust certificates you created or when you are sure of the source of the PDF. The first step is to click on the signature field to bring up the dialog box shown. Your dialog boxes may differ slightly in options depending on the version of Adobe Reader you are using. These examples use Adobe Reader version 9. Note this document is signed so you can follow the steps below for this PDF if you like.

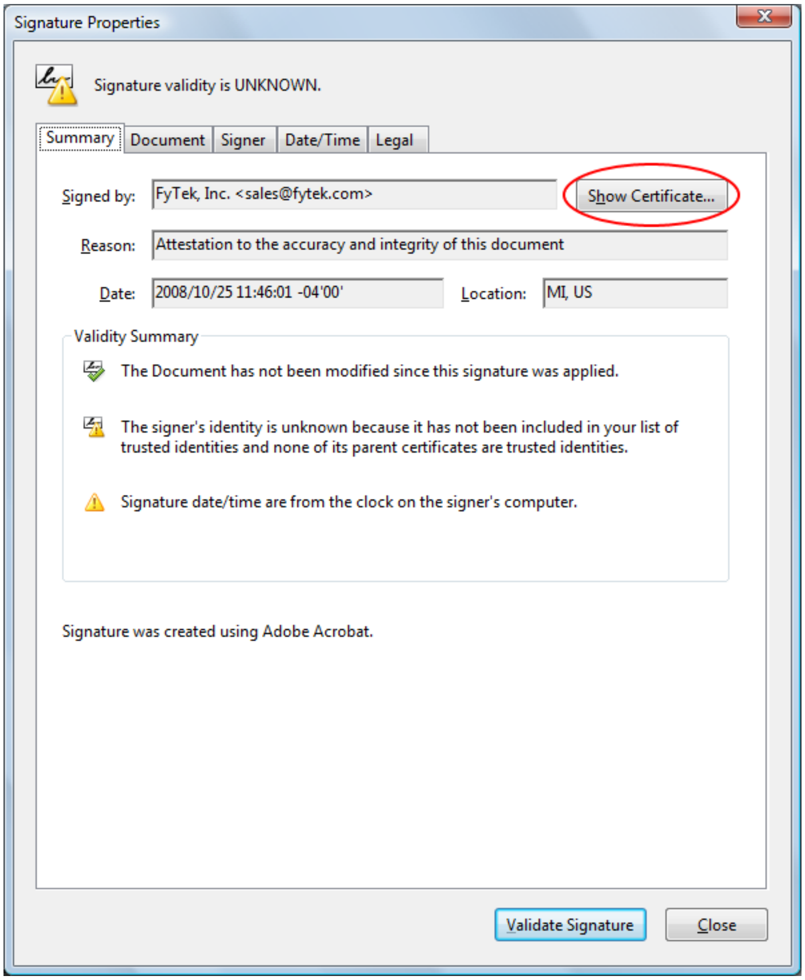
Digital Signatures



Signature Validation Status

Signature validity is UNKNOWN.
- The Document has not been modified since this signature was applied.
- The signer's identity is unknown because it has not been included in your list of trusted identities and none of its parent certificates are trusted identities.

This is the dialog box that appears once you click the signature field. Click the "Signature Properties..." button to continue.



Signature Properties

Signature validity is UNKNOWN.

Signed by: FyTek, Inc. <sales@fytek.com>

Reason: Attestation to the accuracy and integrity of this document

Date: 2008/10/25 11:46:01 -04'00' Location: MI, US

Validity Summary

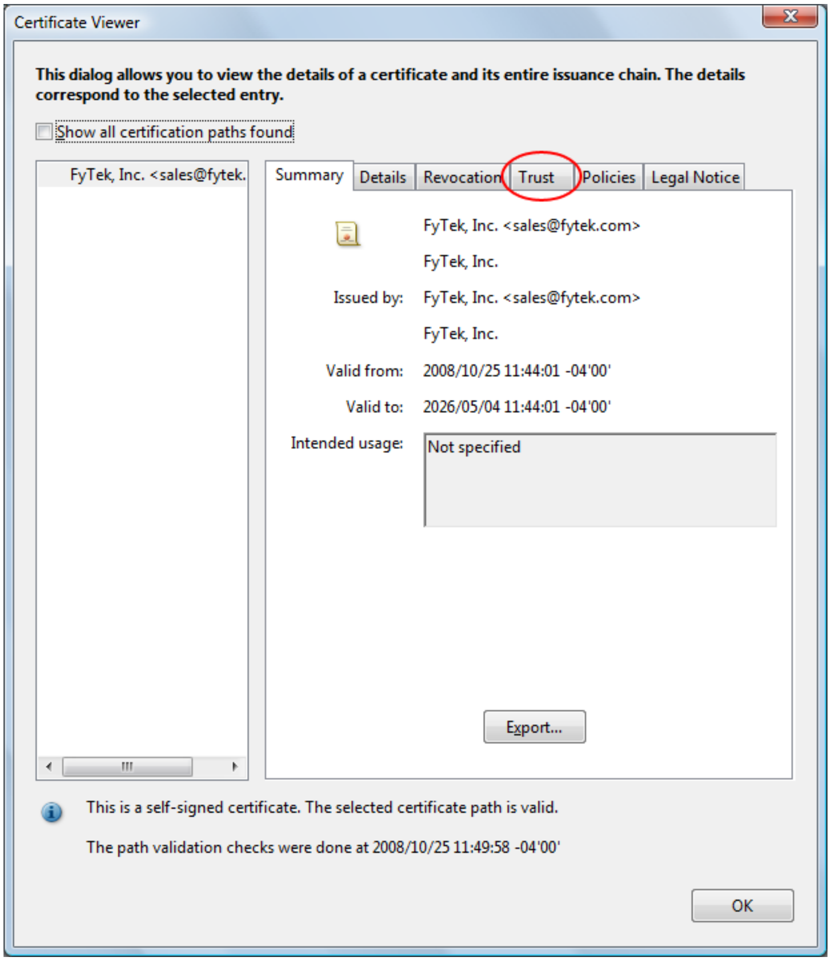
The Document has not been modified since this signature was applied.

The signer's identity is unknown because it has not been included in your list of trusted identities and none of its parent certificates are trusted identities.

Signature date/time are from the clock on the signer's computer.

Signature was created using Adobe Acrobat.

This is the signature property dialog for the certificate. Across the top of the dialog area is a set of tabs you can click on to view various information. For now, click the "Show Certificate..." button to continue.



This dialog allows you to view the details of a certificate and its entire issuance chain. The details correspond to the selected entry.

□ Show all certification paths found

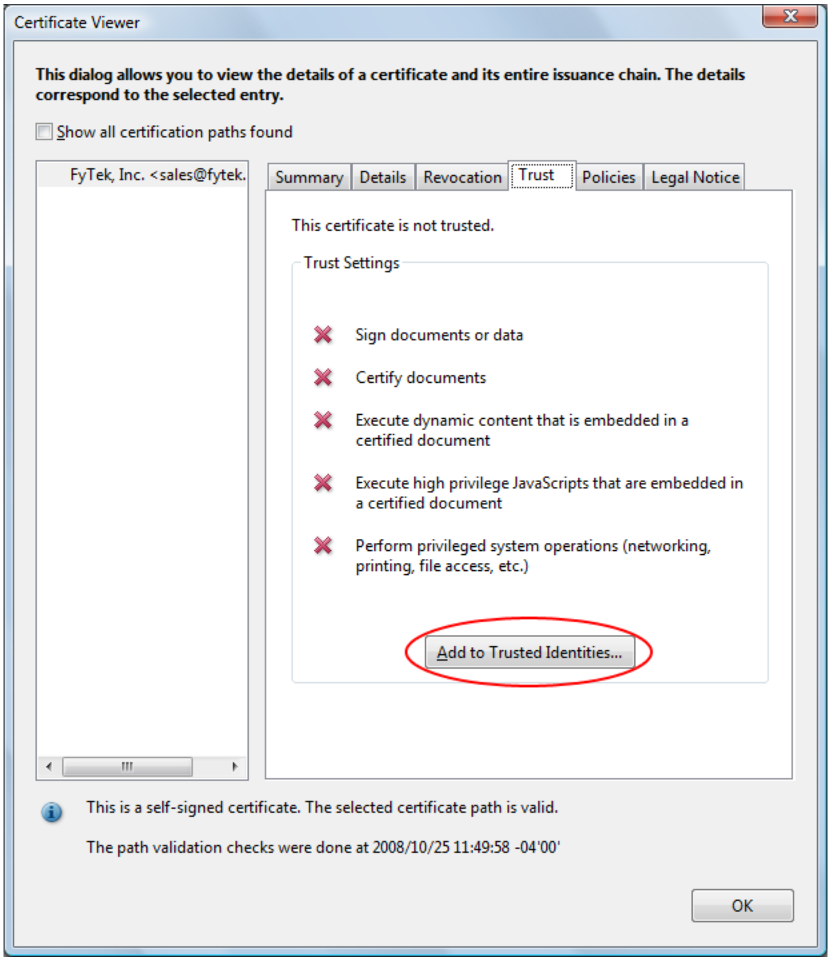
FyTek, Inc. <sales@fytek.

FyTek, Inc. <sales@fytek.com>
FyTek, Inc.
Issued by: FyTek, Inc. <sales@fytek.com>
FyTek, Inc.
Valid from: 2008/10/25 11:44:01 -04'00'
Valid to: 2026/05/04 11:44:01 -04'00'
Intended usage: Not specified

This is a self-signed certificate. The selected certificate path is valid.

The path validation checks were done at 2008/10/25 11:49:58 -04'00'

Another dialog box will open containing a set of tabs across the top. Click on the "Trust" tab.



Certificate Viewer

This dialog allows you to view the details of a certificate and its entire issuance chain. The details correspond to the selected entry.

□ Show all certification paths found

This certificate is not trusted.

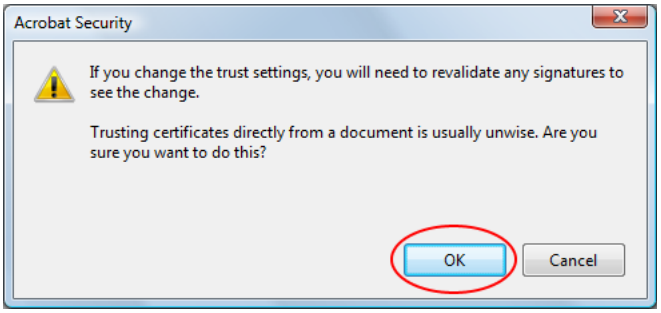
Trust Settings

- ✗ Sign documents or data
- ✗ Certify documents
- ✗ Execute dynamic content that is embedded in a certified document
- ✗ Execute high privilege JavaScripts that are embedded in a certified document
- ✗ Perform privileged system operations (networking, printing, file access, etc.)

This is a self-signed certificate. The selected certificate path is valid.

The path validation checks were done at 2008/10/25 11:49:58 -04'00'

The trust tab shows what trusts you have enabled for the certificate. In this case, no trusts have been established. To trust this certificate, click the "Add to Trusted Identities..." button.



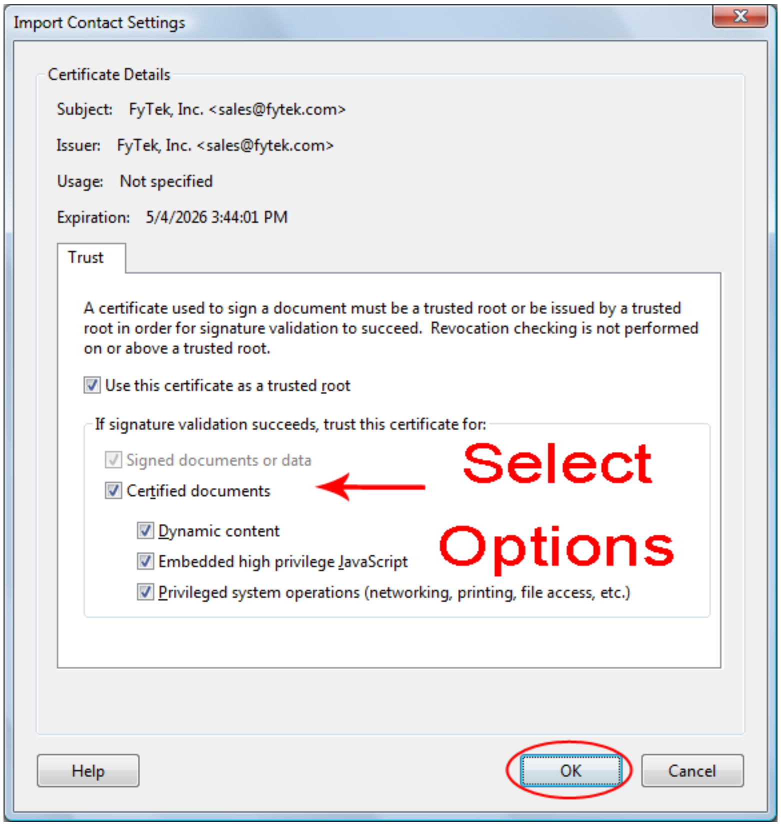
Acrobat Security

If you change the trust settings, you will need to revalidate any signatures to see the change.

Trusting certificates directly from a document is usually unwise. Are you sure you want to do this?

You will likely recieve a warning box. Be sure to only trust certificates when you are certain of their source.

Digital Signatures



Import Contact Settings

Certificate Details

Subject: FyTek, Inc. <sales@fytek.com>

Issuer: FyTek, Inc. <sales@fytek.com>

Usage: Not specified

Expiration: 5/4/2026 3:44:01 PM

Trust

A certificate used to sign a document must be a trusted root or be issued by a trusted root in order for signature validation to succeed. Revocation checking is not performed on or above a trusted root.

☑ Use this certificate as a trusted root

If signature validation succeeds, trust this certificate for:

☑ Signed documents or data

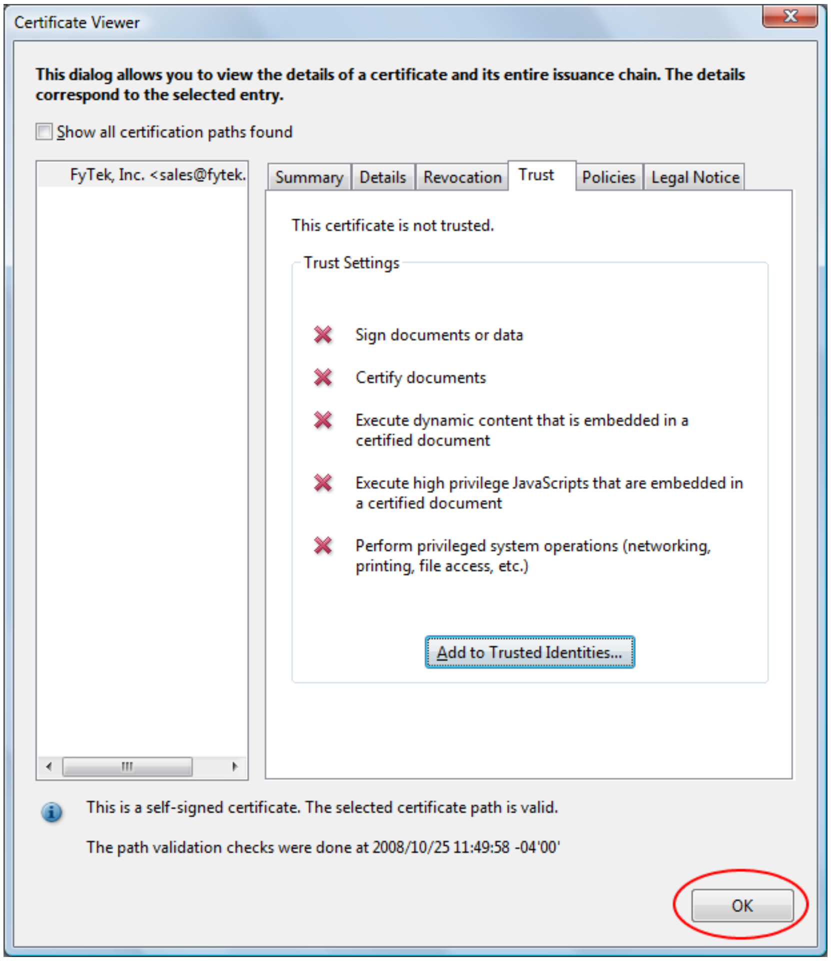
☑ Certified documents

☑ Dynamic content

☑ Embedded high privilege JavaScript

☑ Privileged system operations (networking, printing, file access, etc.)

Select what items you want to trust the certificate for by clicking the checkboxes.



Certificate Viewer

This dialog allows you to view the details of a certificate and its entire issuance chain. The details correspond to the selected entry.

□ Show all certification paths found

FyTek, Inc. <sales@fytek.

This certificate is not trusted.

Trust Settings

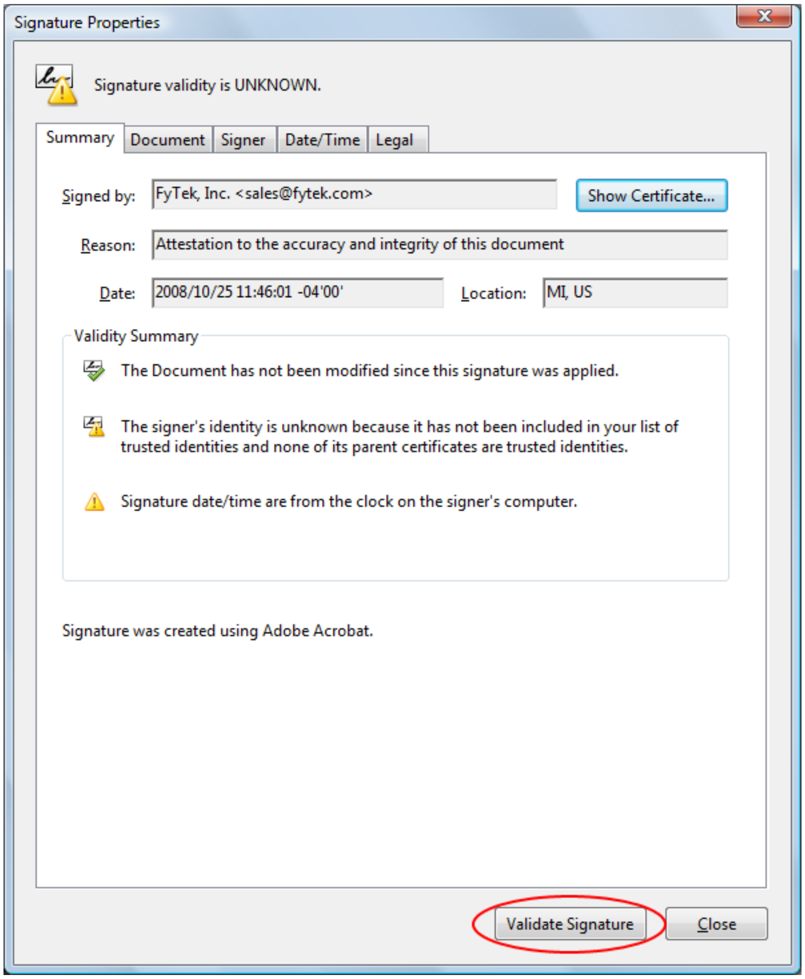
- ✗ Sign documents or data
- ✗ Certify documents
- ✗ Execute dynamic content that is embedded in a certified document
- ✗ Execute high privilege JavaScripts that are embedded in a certified document
- ✗ Perform privileged system operations (networking, printing, file access, etc.)

This is a self-signed certificate. The selected certificate path is valid.

The path validation checks were done at 2008/10/25 11:49:58 -04'00'

Click the "OK" button to continue. Note the red X's will remain until we revalidate the signatures.

Digital Signatures



Signature validity is UNKNOWN.

Signed by: FyTek, Inc. <sales@fytek.com>

Reason: Attestation to the accuracy and integrity of this document

Date: 2008/10/25 11:46:01 -04'00'

Location: MI, US

Validity Summary

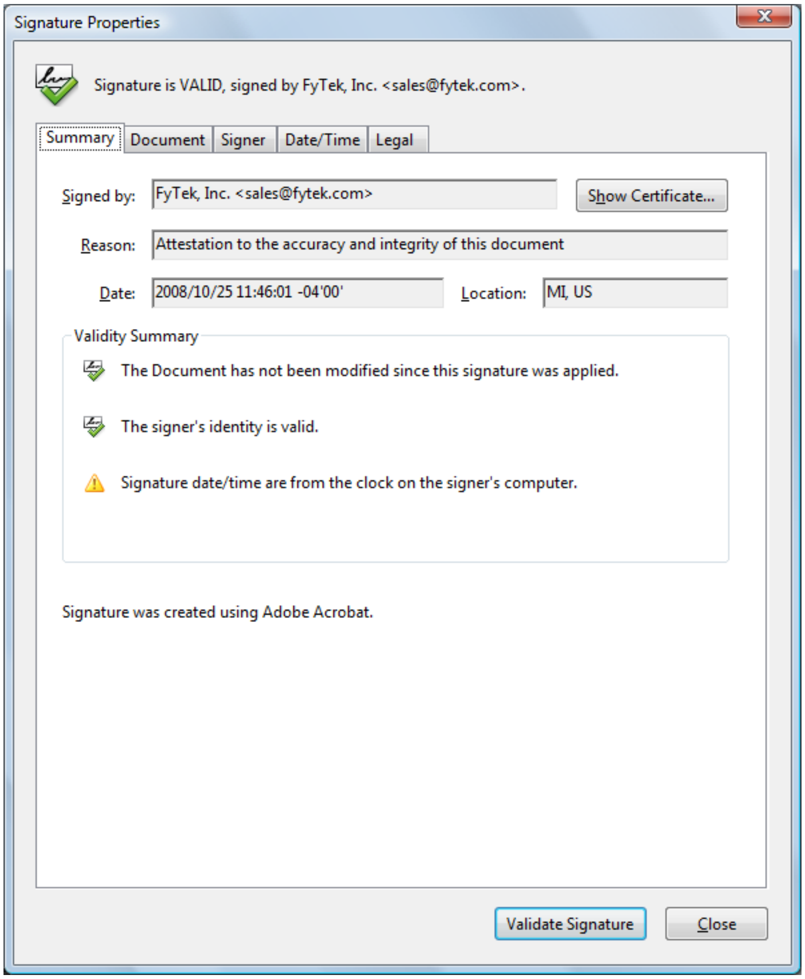
The Document has not been modified since this signature was applied.

The signer's identity is unknown because it has not been included in your list of trusted identities and none of its parent certificates are trusted identities.

Signature date/time are from the clock on the signer's computer.

Signature was created using Adobe Acrobat.

Click the "Validate Signatures" button to validate the signature we just setup the trusts for.



Signature Properties

Signature is VALID, signed by FyTek, Inc. <sales@fytek.com>.

Signed by: FyTek, Inc. <sales@fytek.com>

Reason: Attestation to the accuracy and integrity of this document

Date: 2008/10/25 11:46:01 -04'00'

Location: MI, US

Validity Summary

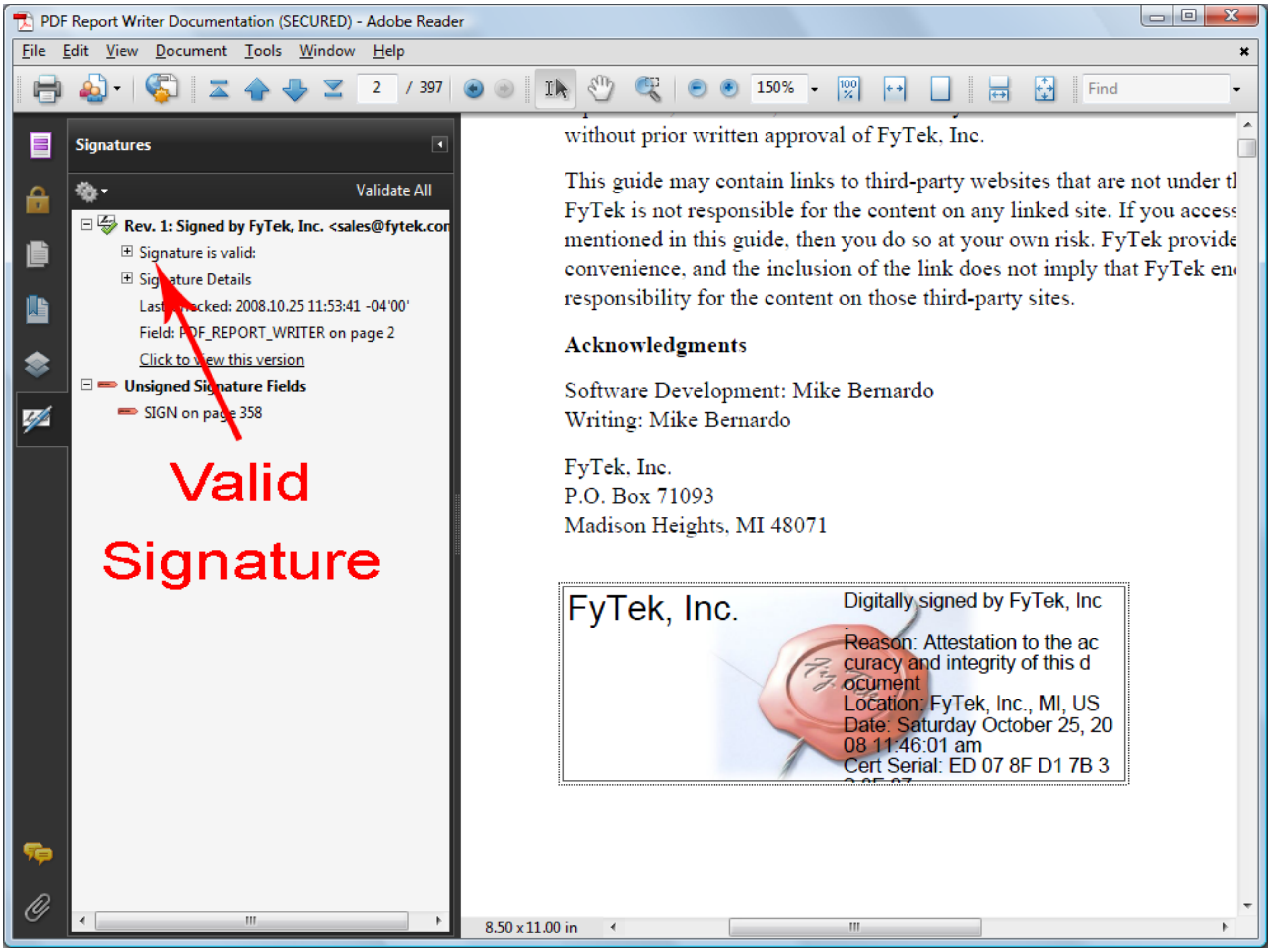
✓ The Document has not been modified since this signature was applied.

✓ The signer's identity is valid.

⚠ Signature date/time are from the clock on the signer's computer.

Signature was created using Adobe Acrobat.

A green check icon now shows in the signature properties dialog.


PDF Report Writer Documentation (SECURED) - Adobe Reader
File Edit View Document Tools Window Help
2 / 397
150%
Find
Signatures
Validate All
8.50 x 11.00 in

Rev. 1: Signed by FyTek, Inc. <sales@fytek.com

Signature is valid:

Signature Details

Last Checked: 2008.10.25 11:53:41 -04'00'

Field: PDF_REPORT_WRITER on page 2

Click to view this version

Unsigned Signature Fields

SIGN on page 358

Valid

Signature

without prior written approval of FyTek, Inc.

This guide may contain links to third-party websites that are not under t FyTek is not responsible for the content on any linked site. If you access mentioned in this guide, then you do so at your own risk. FyTek provide convenience, and the inclusion of the link does not imply that FyTek en responsibility for the content on those third-party sites.

Acknowledgments

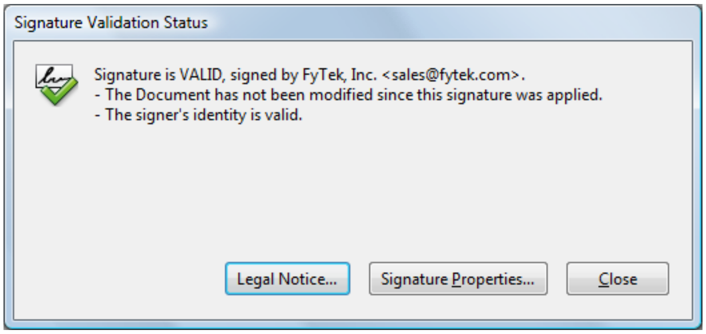
Software Development: Mike Bernardo
Writing: Mike Bernardo

FyTek, Inc.
P.O. Box 71093
Madison Heights, MI 48071

FyTek, Inc.

Digitally signed by FyTek, Inc
Reason: Attestation to the accuracy and integrity of this document
Location: FyTek, Inc., MI, US
Date: Saturday October 25, 2008 11:46:01 am
Cert Serial: ED 07 8F D1 7B 3

A green check icon also shows in the signature pane. All future signings using this certificate will be trusted. The signature pane on the left will show what signings have taken place on the document and what signatures are open for signing. In this case there is one signature so far but an open signature box remains. You may also follow through on the dialog boxes by clicking the second signature (once signed) to view any changes to the document that happened between the time of the first and second signature.



Signature Validation Status

Signature is VALID, signed by FyTek, Inc. <sales@fytek.com>.
- The Document has not been modified since this signature was applied.
- The signer's identity is valid.

This is what you will see now when you first click the signature field, assuming the signature is valid and the PDF has not been tampered with.

Encryption

You can encrypt your PDFs for added security. An encrypted PDF will prompt the user for a password before opening the document using Acrobat or Adobe Reader. There are two passwords that can be set. The owner password and the user password. Opening a document with the owner password allows full access to the document. Opening a document with the user password will place restrictions on what the user can do based on flag settings when the document was created. These restrictions are:

- Disable printing of the document
- Disable changes to the document
- Disable copying of text and/or graphics from the document
- Disable add/change of form fields or annotations

Supplying just an owner password will protect the document against the restrictions above, however no password will be prompted for. No one will be able to print the document if you disable printing and use only an owner password. Supplying just a user password will have the same effect except the password will be prompted for before the document can be opened.

See the list of parameters for the exe version of Report Writer or the methods for the DLL version on how to set these restrictions. You may also use the ENCRYPT tag. The actual contents of the PDF are encrypted in addition to just placing password protection on the document. Standard encryption is 40-bit using a combination of the MD5 and RC4 algorithms. You can specify 128-bit encryption but only users of Acrobat or Adobe Reader 5.0 or higher will be able to open the documents.

XML Data Merge

XML Data Merge is not available with PDF Report Writer SE.

You can use an XML approach to separate your data from the actual layout. This method allows you export your data in XML format and have multiple forms or layouts for presenting that data. You can also modify the layout template without changing the program that pulls data from your database.

The basic concept is to have two files that are merged when creating the PDF - one with your data and one with the layout. The layout has placeholders for the data which are populated from the data file. The DATASET command is used in both files to specify a block of data fields (in the data file) or the area in the layout the data belongs. DATASETs may be nested inside of either files.

Here is a sample data file (note you may also pass in standard XML formatted files):

```
<DATASET ID=main>
  <COMP>Test Company</COMP>
  <DATASET ID=dtl>
    <DATE>1/15/2008</DATE>
    <AMOUNT>2,201.25</AMOUNT>
    <DATASET ID=amts>
      <DESCR>Down Payment</DESCR>
      <AMOUNT>1,201.25</AMOUNT>
    </DATASET>
    <DATASET ID=amts>
      <DESCR>Balance</DESCR>
      <AMOUNT>1,000.00</AMOUNT>
    </DATASET>
  </DATASET>
  <DATASET ID=dtl>
    <DATE>3/15/2008</DATE>
    <AMOUNT>525.31</AMOUNT>
  </DATASET>
</DATASET>
<DATASET ID=main>
  <COMP>XYZ Corporation</COMP>
  <DATASET ID=dtl>
    <DATE>1/31/2008</DATE>
    <AMOUNT>10,321.48</AMOUNT>
  </DATASET>
  <DATASET ID=dtl>
    <DATE>2/20/2008</DATE>
    <AMOUNT>1,495.28</AMOUNT>
  </DATASET>
  <DATASET ID=dtl>
    <DATE>3/31/2008</DATE>
    <AMOUNT>986.57</AMOUNT>
  </DATASET>
</DATASET>
```

A sample layout file:

```
<PDF>
<DATASET ID=main>
<PAGE>
<HEADER MARGIN=10>
<TEXT ALIGN=C FACE=16 SIZE=18>
<&COMP>
</TEXT>
</HEADER>
<BODY>
<TABLE BORDER=1 CELLPADDING=5>
<TH ALIGN=C BGCOLOR=blue FCOLOR=white>
<TD>Date</TD>
<TD>Amount</TD>
</TH>
<DATASET ID=dtl>
<TR>
<TD><&DATE></TD>
<TD ALIGN=R><&AMOUNT></TD>
</TR>
<DATASET ID=amts>
<FIRSTOF ID=amts>
<TR><TD></TD><TD>
<TABLE BORDER=1 CELLPADDING=5>
<TH ALIGN=C BGCOLOR=blue FCOLOR=white>
<TD>Description</TD>
<TD>Amount</TD>
</TH>
</FIRSTOF>
<TR>
<TD><&DESCR></TD>
<TD ALIGN=R><&AMOUNT></TD>
</TR>
<LASTOF ID=amts>
</TABLE>
</TD></TR>
</LASTOF>
</DATASET>
</DATASET>
</TABLE>
</BODY>
</DATASET>
```

Variables are placed in the layout as a tag with an & in front of the variable name. The same technique is used in the data file except the & is left off of the tag. There is also a closing tag in the data file for each variable. The data between the opening and closing tag in the data file is the value that will go into the placeholder in the layout.

The variables in the layout may also contain a list of replacements. This can be useful in cases where you do not have control over the XML data and need to make minor adjustments before rendering. In this case use the REPLACE="text" option. The "text" is a comma separated list of strings to change from and to. If you are using the comma as a replacement string then use the SEP="text" option to specify a different separator. For example, <&somevar REPLACE=",|.|dev|test" SEP="|"> will replace any comma with a period and the string "dev" with "test" before rendering the value for "somevar".

There are three levels of data in the example above but you can nest more levels if you need to. The first level, using DATASET with ID=main, is the header information. The second, using DATASET with ID=dtl, is the detail information. The third, using DATASET with ID=amts, is the amount detail information used for one of the detail blocks. You pass the layout and data on the command line as follows:

pdfrw.exe filein.frw fileout.pdf -data data.xml

Where filein.frw is the layout, fileout.pdf is the output file and data.xml is the data file. The DLL uses the method setDataFile or setDataCmd to specify the data file or data commands to use.

You may also modify the Report Writer commands that are used based on the data. The <PERL> and </PERL> tags are used to denote a block of Perl code that uses the data values in conditional statements. The -allowperl option or setAllowPerl method must be specified when executing Report Writer for the Perl code to be processed. For example, using the above layout file, here is a modification that will change the text color for the AMOUNT field when the company name begins with XYZ. Replace the <TD ALIGN=R> in front of the <&AMOUNT> with the following:

```
<PERL>
if (substr("<&COMP>",0,3) eq "XYZ")
{
  return "<TD ALIGN=R FCOLOR=red>";
}
else
{
  return "<TD ALIGN=R>";
}
</PERL>
```

The 'return "...";' statement contains the commands that will be used. In the case above, when the value of &COMP starts with XYZ the first return statement will be used. For all other cases, the second return statement is used. Use the -e option or setErrFile method when debugging the Perl code. Any errors Perl generates from the command will be placed in the error log.

The <NOPERL> and </NOPERL> tags allow placement of default Report Writer tag(s) when not parsing Perl code (when the -allowperl or setAllowPerl has not been used). These tags go inside of the <PERL> and </PERL> tags. Here's another example of the above where a default TD tag is used when not parsing Perl code:

```
<PERL>
<NOPERL>
<TD ALIGN=R>
</NOPERL>
if (substr("<&COMP>",0,3) eq "ABC")
{
  return "<TD ALIGN=R FCOLOR=red>";
}
else
{
  return "<TD ALIGN=R>";
}
</PERL>
```

Note in this case there is no return statement or quotes around the text returned for NOPERL. Everything outside of the NOPERL block (but still within the PERL block) is ignored in this case.

Here's a final example using a Perl subroutine making use of the INCLUDE option:

```
<PERL INCLUDE>
sub chkcomp
{
  my @parms = @_;
  my $ret = "";
  if (substr($parms[0],0,3) eq "ABC")
  {
    $ret = "<TD ALIGN=R FCOLOR=red>";
  }
  else
  {
    $ret = "<TD ALIGN=R>";
  }
  return $ret;
}
</PERL>
. . .
<PERL>
<NOPERL>
<TD ALIGN=R>
</NOPERL>
return chkcomp("<&COMP>");
</PERL>
```

The *chkcomp* function is available to all PERL blocks where the INCLUDE option isn't used. Also note that all variables (@parms and $ret) are declared using the "my" keyword.

Alternatively, you may pass in standard XML files that start with the tag <?xml version="1.0"?>. This file will be pre-processed and converted to the DATASET layout described above. Your file should be ISO-8859-1 or UTF-8 encoded. All UTF-8 encoded XML files will be converted to ISO-8859-1. Any characters above 255 (single-byte characters range from 0 to 255) in UTF-8 files will not be converted.

The -dataout option or setDataFileOut method can be used to write the contents of the converted XML input to a file on disk. It may be helpful to view the converted file to see where the DATASET tags are placed and what the converted file looks like.

The DATASET levels are created based on how the data is grouped in the XML file. Note the root element is the top most DATASET. The ID value for the DATASET is taken from the tag name. Parent elements in the source XML convert to DATSETs. Child elements and attributes become name/value pairs.

Pass a comma separated list of attribute names to further break them out into DATASETS. For example:

```
<?xml version="1.0"?>
<note>
<msg num="1">
<from>Kris</from>
<to>Joe</to>
</msg>
<msg num="2">
<from>Jane</from>
<to>Tom</to>
</msg>
</note>
```

Is converted to:

```
<DATASET ID=note>
   <DATASET ID=msg>
     <to>Joe</to>
     <num>1</num>
     <from>Kris</from>
   </DATASET>
   <DATASET ID=msg>
     <to>Tom</to>
     <num>2</num>
     <from>Jane</from>
   </DATASET>
</DATASET>
```

The -datakeyattr option or setDataFileKeyAttr method can be used to further break on the "num" attribute of "msg". They take a value or a string of comma separated values to further break on. Any attributes matching the string passed become their own DATASET block. By passing the string "num" to this option or method the above example becomes:

```
<DATASET ID=note>
   <DATASET ID=msg>
     <DATASET ID=1>
       <to>Joe</to>
       <from>Kris</from>
     </DATASET>
     <DATASET ID=2>
       <to>Tom</to>
       <from>Jane</from>
     </DATASET>
   </DATASET>
</DATASET>
```

Note there are now DATASETs within a single "msg" block instead of two "msg" blocks. You'll want to set these options depending on how you want the data arranged.

You may use an element=>id syntax instead if you want more control over which elements and attributes are broken out. For example, suppose you have another tag called "note" that also has a "num" attribute but you don't want to break out a DATASET for this element. You would use "msg=>num" rather than just "num" in this case. Only the "msg" elements will be broken out on "num" and not the "note" elements. A comma may be used to separate multiple entries. Either specify just a list of attributes to break out or a list of element/attribute pairs but don't mix the two. For example, "msg=>num,type" is not valid.

The layout file (the .frw) does not change based on the type of input XML file used. It still uses the DATASET tags and the <&var> syntax for variable values.

Functions

A function lets you perform a calculation that might otherwise be difficult to do depending on your data source. Note you must use the -allowperl option or setAllowPerl method when using functions. See the file sample18.frw for an example of using functions.

Functions are used within the RWGET tag to perform some mathematical function on a cell or group of cells. First, use the NAME="text" option on a TD (table cell) tag to name a cell or range of cells. For example, <TD NAME="price">. Next, reference the NAME in the RWGET tag using the fn(opt:'value',...) function. For example, <TD><RWGET fn(name:'price', type:'sum')><TD> will add up the contents of all cells with the name "price" and display that value for the contents of the table cell. All of the options passed to fn should be in the form option name followed by a colon then the option value in quotes. The value of the option must be in quotes for it to be recognized. The fn function takes the following options:

Parameter	Description
NAME	The name of the group of cells you wish to reference. For example, name:'price'.
NAME2	The name of a second group of cells when using the weighted average function.
TABLE	Optional. The case-sensitive name of the table you wish to reference. The default is the current table. Use the NAME option on the TABLE tag when creating the table and supply that value here. For example, table:'base'.
FORCENUM	Optional. Supply if the cell contents for the NAME group contain formatting such as $ or commas. This will strip out the extraneous characters and treat the result as a number. You do not need to pass a value with this option. Note that this is assumed when TYPE is set to AVG, SUB, SUM, or WAVG.
ROW	Optional. If not specified, all cells with the matching NAME are included. Pass a number to refer to particular row in the table. Rows are numbered sequentially starting from 1 from the first TR row. You may also pass a + or - in front of the number to mean use the value as a relative offset from the current row number. Using row:'+0' would mean the current row.
THISROW	Optional. You do not need to pass a value with this option. Means the same as having row:'+0'.

Functions

Parameter	Description
TYPE	The function to apply. Pass one of the following: AVG - The average value MAX - The maximum value MIN - The minimum value SUB - Sum selected values SUM - Sum the values WAVG -The weighted average (use both NAME and NAME2) If left blank, the last entry for a cell with the specified name is returned.

The SUB function provides a quick way of producing a set of sub-totals. It includes only the matching cells that are less than the current row number. Additionally, the function remembers where it left off so you can place as many sub-totals in your report as you need.

You may wrap the fn() function inside of another function called numformat. Call the function like this: numformat(x, mask [,dec, thousands sep]). In this case, x is the number to format and mask is the format mask. Optionally include the decimal character and thousands separator which would be the comma and period for European formats. Use the # symbol for numbers in the mask and be sure to make the mask large enough for the biggest number that might be returned. For example, numformat(1234.5678,'#,###.##') would result in 1,234.57 being displayed. Along with the fn() function, you might have something like this: <RWGET numformat(fn(name:'price', type:'sum'), '##,###.##')>

In addition to a single function call, you can also perform other mathematical functions as well. For instance, you can use base functions for addition (+), subtraction (-), multiplication (*), and division (/). Use a period (.) for string concatenation. To compute a 6% tax you might use fn(name:'price', type:'sum') * .06. You may also use mutiple functions in the same RWGET such as fn(name:'orange',type:'sum') + fn(name:'apple', type:'sum'). The RWSET function can save the result to a variable to reference later if you need to use the function in multiple locations. For example, <RWSET $tot = fn(name:'orange', type:'sum') + fn(name:'apple', type:'sum')>. Then, simply use $tot in RWGET statements to reference the sum.

Use the row option or "thisrow" keyword to perform a calculation on a value from a single row. For instance, assume you have a row like this:
<TD NAME='qty'><TD NAME='price'>
You could use fn(name:'qty', thisrow) * fn(name:'price', thisrow) to multiple quantity times price. Additionally, you may use any Perl functions or operators on these values. Treat the fn() as a simple value in any equation, "if" statement or any other processing you might want to perform.

Functions

Use the RWFUNCT tag to store commonly used functions. The function "funct" is a special function that will be replaced by the text defined using the RWFUNCT tag. For example, funct(calc) where calc is defined with RWFUNCT by setting NAME="calc".

XPS Documents

XPS Document Output is not available with PDF Report Writer SE.

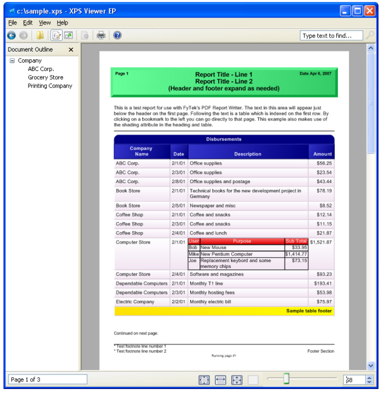
XPS document format is a paginated-document specification developed by Microsoft. This format is similar to PDF in the sense it is a finalized output not intended to be edited, unlike a document saved from a word processor where you can re-open and perform text or layout edits.

Not all features found in PDF are available in XPS and not all Report Writer options have been implemented. The following is a list of items which are available:

- Most text output except changing font color in the middle of a string
- Embedded fonts
- Most images
- Line/Rectangle drawing
- Tables along with most formatting such as borders, cell background colors and so forth.
- Bookmarks and links (though hyperlink text is not colored blue)

You may use the XPSPAGE tag to place a background on your XPS output. This is similar to the PDFPAGE tag for PDFs. The XPSPAGE option is ignored for PDFs and similarly, PDFPAGE is ignored for XPS files.

If you have an existing PDF you want to use as an XPS background, the easiest way to convert it is to simply print the PDF as an XPS document. You'll need to install Microsoft's XPS Document Writer on XP based systems first. Simply open the PDF in Reader, select Print and choose the XPS Document Writer as the output. You'll be prompted for a file name to save as. Use this file along with the XPSPAGE option.



Page 1 Report Title - Line 1 Date Apr 6, 2007
Report Title - Line 2
(Header and footer expand as needed)

This is a test report for use with FyTek's PDF Report Writer. The text in this area will appear just below the header on the first page. Following the text is a table which is indexed on the first row. By clicking on a bookmark to the left you can go directly to that page. This example also makes use of the shading attribute in the heading and table.

Disbursements			
Company Name	Date	Description	Amount
ABC Corp.	2/1/01	Office supplies	$56.25
ABC Corp.	2/3/01	Office supplies	$23.54
ABC Corp.	2/8/01	Office supplies and postage	$43.44
Book Store	2/1/01	Technical books for the new development project in Germany	$78.19
Book Store	2/5/01	Newspaper and misc	$8.52
Coffee Shop	2/1/01	Coffee and snacks	$12.14
Coffee Shop	2/3/01	Coffee and snacks	$11.15
Coffee Shop	2/4/01	Coffee and lunch	$21.87
Computer Store	2/1/01	User / Purpose / Sub Total Bob / New Mouse / $33.95 Mike / New Pentium Computer / $1,414.77 Joe / Replacement keybord and some memory chips / $73.15	$1,521.87
Computer Store	2/4/01	Software and magazines	$93.23
Dependable Computers	2/1/01	Monthly T1 line	$193.41
Dependable Computers	2/3/01	Monthly hosting fees	$53.98
Electric Company	2/2/01	Monthly electric bill	$75.97
Sample table footer			

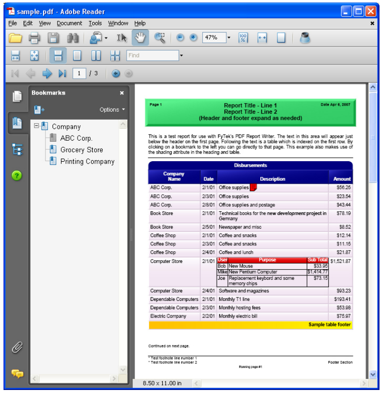
Continued on next page.

* Test footnote line number 1
* Test footnote line number 2

Footer Section

Running page #1

XPS Document



Page 1 Report Title - Line 1 Date Apr 6, 2007
Report Title - Line 2
(Header and footer expand as needed)

This is a test report for use with FyTek's PDF Report Writer. The text in this area will appear just below the header on the first page. Following the text is a table which is indexed on the first row. By clicking on a bookmark to the left you can go directly to that page. This example also makes use of the shading attribute in the heading and table.

Disbursements ... Sample table footer

Continued on next page.

Footer Section

Running page #1

PDF Document

XPS Document files may be larger than PDFs in some cases, especially when the PDF doesn't contain any added images or fonts. The reason is the

PDF viewer contains 14 built-in fonts so those do not need to be included in the PDF. XPS format, however, requires that fonts or subsets of fonts be included in all cases. The inclusion of the fonts is what makes some XPS files larger than PDF.

Note these are the features currently supported. Additional features will be added over time based on customer needs.

Backgrounds

PDF Report Writer has two features for creating backgrounds. The first is the combination of the NEWFORM option and USEFORM tag. This method requires you to set up the background, which could be nothing more than an image, a set of line/text commands or combination of the two. The NEWFORM option on the PAGE tag tells the software to store the following page for use as a background but don't include the background page by itself as a page in the PDF. The header and footer on all pages in this document are using this method.

Another option for setting backgrounds is to use one or more existing PDFs and reference a page or combination of pages as backgrounds. You should use a relatively small PDFs for this function, preferably ones with just the page(s) you are going to use as backgrounds. The larger the PDF, the more time it will take to parse though and map out the details. The PDFs do not have to be ones created using PDF Report Writer. Not all existing PDFs will work - encrypted PDFs are not supported for this feature, nor are PDFs that have been modified. You may use encryption on the resulting PDF which will also encrypt the contents of the original PDFs. Easiest way to tell if your existing PDFs will work is to just try it. Use the PDFPAGE command to specify which pages you want to use and the source PDF file.

Note that the bookmark structure from the original PDFs will not be retained. Only the content of the original page are kept. Any bookmarks you place on the background will work just the same as when not using a background. Also see the PULLPAGE command for another way of inserting pages from an existing Report Writer PDF.

Example: To use page 3 of an existing PDF with the executable version of the program run the following:

```
pdfrw.exe myfile.frw myfile.pdf -pdf backgrd.pdf
```

where myfile.frw is the file to convert to PDF, myfile.pdf is the output and backgrd.pdf is an existing PDF to use pages from. Inside of myfile.frw, make a reference to the page from the existing PDF with the PDFPAGE command like this:

```
<PDF>
<PAGE>
<GRID X=.5 Y=.5 XUNITS=80 YUNITS=80>
<PDFPAGE VALUE=3>
<TEXT X=5 Y=10 SIZE=12>
Here is some text to print on top of this background.
</TEXT>
<TEXT X=15 Y=20 SIZE=10>
Here is some more text.
</TEXT>
```

Note the use of the X and Y positioning on the TEXT statements. Use the X and Y, and possibly the X2 and Y2 options, to position the text where you want on the page. Use the FORCE option on the TEXT as well if you want to make sure the text doesn't wrap.

Here is an example of pulling in pages from serveral PDFs.

```
<PDF>
<PAGE>
<GRID X=.5 Y=.5 XUNITS=80 YUNITS=80>
<PDFPAGE SRC="pdf1.pdf" VALUE=1>
<TEXT X=5 Y=10 SIZE=12>
Here is some text to print on top of this background.
</TEXT>
<TEXT X=15 Y=20 SIZE=10>
Here is some more text.
</TEXT>
<PAGE>
<PDFPAGE SRC="pdf2.pdf" VALUE=1>
<TEXT X=5 Y=10 SIZE=12>
Here is some text to print on top of this background.
</TEXT>
<PAGE>
<PDFPAGE SRC="pdf2.pdf" VALUE=2>
<TEXT X=5 Y=10 SIZE=12>
Here is some text to print on top of this background.
</TEXT>
<PAGE>
<PDFPAGE SRC="pdf2.pdf" VALUE=3>
<PDFPAGE SRC="pdf3.pdf" VALUE=1>
<TEXT X=5 Y=10 SIZE=12>
Here is some text to print on top of this background.
Two pages from two different PDFs have been overlaid in this case.
</TEXT>
```

Colors

Parameters such as FCOLOR, SCOLOR and BORDERCOLOR take a color value. Colors may be entered in any of the following ways:

- You may specify the red, green and blue components as decimal values from 0 to 1, separated by a comma. In this case 0,0,0 is black and 1,1,1 is white.
- You may specify the red, green and blue components as values from 0 to 255, separated by a comma. In this case 0,0,0 is black and 255,255,255 is white.
- You may specify the red, green and blue components as a hex string preceded by a # sign. In this case #000000 is black and #FFFFFF is white. If all three red, green and blue components are pairs of the same character, such as #ee33dd, you may shorten to #e3d. When three characters are found after the # sign they are expanded by duplicating each chracter to make the longer six character code.
- You may specify the cyan, magenta, yellow and black components (CMYK) as decimal values from 0 to 1, separated by a comma. In this case 0,0,0,1 or 1,1,1,0 is black and 0,0,0,0 is white.
- You may specify the cyan, magenta, yellow and black components (CMYK) as decimal values from 0 to 1, separated by a comma. In this case 0,0,0,255 or 255,255,255,0 is black and 0,0,0,0 is white.
- You may specify one of the RGB colors in the table below or use the Pantone® color chart on the following page. (Pantone is a registered trademark of Pantone, Inc.)

Color	Name
⬛	Black
░	Silver
▒	Gray
⬜	White
🟫	Maroon
🟥	Red
🟪	Purple
🟪	Fuchsia

Color	Name
🟩	Green
🟩	Lime
🟫	Olive
🟨	Yellow
🟦	Navy
🟦	Blue
🟦	Teal
🟦	Aqua

Colors

Use the number or name shown as the color value (enclose names with spaces in quotes). Pantone® colors shown are converted to CMYK values in the PDF.

PROCESS YELLOW	PROCESS MAGENTA	PROCESS CYAN	PROCESS BLACK	100	101	102	PANTONE YELLOW	103	104	105	106	107	108	109	110	111	112	113	114	115
116	117	118	119	120	121	122	123	124	125	126	1205	1215	1225	1235	1245	1255	1265	127	128	129
130	131	132	133	134	135	136	137	138	139	140	1345	1355	1365	1375	1385	1395	1405	141	142	143
144	145	146	147	148	149	150	151	152	153	154	1485	1495	1505	ORANGE 021	1525	1535	1545	155	156	157
158	159	160	161	1555	1565	1575	1585	1595	1605	1615	162	163	164	165	166	167	168	1625	1635	1645
1655	1665	1675	1685	169	170	171	172	173	174	175	176	177	178	WARM RED	179	180	181	1765	1775	1785
1788	1795	1805	1815	1767	1777	1787	RED 032	1797	1807	1817	182	183	184	185	186	187	188	189	190	191
192	193	194	195	1895	1905	1915	1925	1935	1945	1955	169	197	198	199	200	201	202	203	204	205
206	207	208	209	210	211	212	213	214	215	216	217	218	219	RUBINE RED	220	221	222	223	224	225
226	227	228	229	230	231	232	RHODAMINE RED	233	234	235	236	237	238	239	240	241	242	2365	2375	2385
2395	2405	2415	2425	243	244	245	246	247	248	249	250	251	252	PANTONE PURPLE	253	254	255	256	257	258
259	260	261	262	2562	2572	2582	2592	2602	2612	2622	2563	2573	2583	2593	2603	2613	2623	2567	2577	2587
2597	2607	2617	2627	263	264	265	266	267	268	269	2635	2645	2655	2665	VIOLET	2685	2695	270	271	272
273	274	275	276	2705	2715	2725	2735	2745	2755	2765	2706	2716	2726	2736	2746	2756	2766	2707	2717	2727
BLUE 072	2747	2757	2767	2708	2718	2728	2738	2748	2758	2768	277	278	279	REFLEX BLUE	280	281	282	2905	2915	2925
2935	2945	2955	2965	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299
300	301	302	303	2975	2985	2995	3005	3015	3025	3035	304	305	306	PROCESS BLUE	307	308	309	310	311	312
313	314	315	316	3105	3115	3125	3135	3145	3155	3165	317	318	319	320	321	322	323	324	325	326
327	328	329	330	3242	3252	3262	3272	3282	3292	3302	3245	3255	3265	3275	3285	3295	3305	3248	3258	3268
3278	3288	3298	3308	331	332	333	PANTONE GREEN	334	335	336	337	338	339	340	341	342	343	3375	3385	3395
3405	3415	3425	3435	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381
382	383	384	385	386	387	388	389	390	391	392	400	401	402	403	404	405	PANTONE BLACK	406	407	408
409	410	411	412	393	394	395	396	397	398	399	3935	3945	3955	3965	3975	3985	3995	413	414	415
416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436
437	438	439	440	441	442	443	444	445	446	447	WARM GREY 1	WARM GREY 2	WARM GREY 3	WARM GREY 4	WARM GREY 5	WARM GREY 6	WARM GREY 7	WARM GREY 8	WARM GREY 9	WARM GREY 10
WARM GREY 11	COOL GREY 1	COOL GREY 2	COOL GREY 3	COOL GREY 4	COOL GREY 5	COOL GREY 6	COOL GREY 7	COOL GREY 8	COOL GREY 9	COOL GREY 10	COOL GREY 11	448	449	450	451	452	453	454	4485	4495
4505	4515	4525	4535	4545	455	456	457	458	459	460	461	462	463	464	465	466	467	468	4625	4635
4645	4655	4665	4675	4685	469	470	471	472	473	474	475	4695	4705	4715	4725	4735	4745	4755	476	477
478	479	480	481	482	483	484	485	486	487	488	489	504	505	506	507	508	509	510	511	512
513	514	515	516	517	5115	5125	5135	5145	5155	5165	5175	497	498	499	500	501	502	503	4975	4985
4995	5005	5015	5025	5035	490	491	492	493	494	495	496	518	519	520	521	522	523	524	5185	5195
5205	5215	5225	5235	5245	525	526	527	528	529	530	531	5255	5265	5275	5285	5295	5305	5315	532	533
534	535	536	537	538	539	540	541	542	543	544	545	5395	5405	5415	5425	5435	5445	5455	546	547
548	549	550	551	552	5463	5473	5483	5493	5503	5513	5523	5467	5477	5487	5497	5507	5517	5527	553	554
555	556	557	558	559	5535	5545	5555	5565	5575	5585	5595	560	561	562	563	564	565	566	5605	5615
5625	5635	5645	5655	5665	567	568	569	570	571	572	573	574	575	576	577	578	579	580	5743	5753
5763	5773	5783	5793	5803	5747	5757	5767	5777	5787	5797	5807	600	601	602	603	604	605	606	607	608
609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	581	582
583	584	585	586	587	5815	5825	5835	5845	5885	5865	5875	628	629	630	631	632	633	634	635	636
637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657
658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678
679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699
700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720
721	722	723	724	725	726	727	728	729	730	731	732	BLACK 2	BLACK 3	BLACK 4	BLACK 5	BLACK 6	BLACK 7			

Client-Server TCP/IP

Running PDF Report Writer as a server is a way to startup the program and have it remain idle until it receives a request via TCP/IP to build a PDF. Once it completes its request it will process any other waiting requests (unless a pool is specified to allow multiple users at once) until there are no more. The program will then go back into wait mode until another request comes in. The advantage to running PDF Report Writer this way is you bypass the startup time for each run of the program. This may not be an issue if you perform a few builds each day but if you are running hundreds it could add up. In addition, the processing happens on the server so client machines are not using CPU time building reports.

There is also the .NET and COM compatible DLL (pdfrw_20.dll) that can be used to start/stop the server as well as build files. It can also serve as a DLL wrapper to the base program when not using server mode. But it makes starting and stopping the server along with building PDFs easy with some built in methods. The source code is available on GitHub.

To start the server using the .NET DLL set your license info and then start the server (default values are shown for the parameters):

```
licInfo(String licName,
            String licPwd,
            int autoDownload)
startServer(String host = "localhost",
            int port = 7075,
            int pool = 5,
            String log = "")
```

If you don't have a license, use "fytek-inc" and "abc12345" for the name and password. Then, either in the same program or another one, pass the data or files to call and build your output using the method buildReport. You may optionally receive back the PDF as a byte array to the method. See the DLL methods section for more details on the methods available.

There are several programs (note do not include the .exe under Unix) used for running in server mode. They are:
Server Programs
pdfrw.exe (or pdfrw64.exe) - used to start a server from the command line (by passing -server as the first option)

pdfrw_srv.exe (or pdfrw_srv64.exe) - used to install a server as a Windows service

Client Programs

pdfrw_tcp.exe (or pdfrw_tcp64.exe) - used to submit a client request to the server

pdfrw_gui_tcp.exe (or pdfrw_gui_tcp64.exe) - same as pdfrw_tcp.exe but with a progress dialog box

The -server option is used to start up PDF Report Writer in server mode like this:

```
C:\>pdfrw -server -v -pool 5
  -log "c:\logs\rwlog.txt" -host "localhost"
  -port 7075 -licname "fytek-inc" -licpwd "abc12345"
  -licweb
```

This starts the program in a DOS or Unix command session where it will remain until cancelled or a -quit command is sent. The preferred way to run under Windows is to use the PDF Report Writer Service. The program is installed as a Windows service that any user with network access and permission may use. This section contains all the options that apply to both this method and the service.

You can run in the background like this (note the & at the end of the command) on Unix platforms:

```
$ pdfrw -server -v -pool 5
  -log "/logs/rwlog.txt" -host "localhost"
  -port 7075 -licname "fytek-inc" -licpwd "abc12345"
  -licweb &
```

The program will startup and wait for commands on the specified port (7075 is the default if not set). The -server option must be the first option passed to the program. In addition, you must pass in your subscription (-licname and -licpwd options) or server key (-kn and -kc options). You do not need to include the subscription or server key information on client requests.

PDF Report Writer should then start and wait for commands. You issue commands by sending them to the TCP/IP port. Requests will be handled in sequence as they arrive unless the -pool option is used. This may allow for much faster processing as the program is already running in the background waiting for a request rather than starting up a separate process, performing its task, then shutting back down each time.

You may use any program to send the commands to the TCP/IP port in additional to the included pdfrw_20.dll which handles many of these details for you. Pass BUILDPDF followed by a line feed (ASCII 10) to the port to indicate all information has been sent and PDF Report Writer should start processing. Or you may use the included pdfrw_tcp.exe (pdfrw_tcp on Unix) to perform the call to the server. This program will take care of opening the port, sending the parameters you give it and including the BUILDPDF command. The program pdfrw_tcp.exe does not build the output - it simply sends the commands to the port for processing by the server. To use pdfrw_tcp, fisrt start the server as described above. Pass any valid PDF Report Writer commands to pdfrw_tcp and optionally include the -host and -port commands. Here is an example:

```
pdfrw_tcp -host localhost -port 7075
  file1.frw -data file2.xml fileout.pdf
```

If the PDF Report Writer server is running, it will process the command otherwise an error will be returned. Note that the PDF Report Writer server is processing the request so you may need to provide the full path of your input and output files otherwise file names will be relative to the directory where the PDF Report Writer server is running. You can also use the -cwd or -currdir options to change the working directory. Also, the files must be available from the server rather than the client. That is, if you are sending commands from a local Windows client to process on a Linux server, the PDF files must be available on the Linux server (vs. the Windows client) since PDF Report Writer is running on the Linux server in this case. The file pathing in this example should be based on the Linux directory structure and not Windows.

You may wish to send PDFs to the server if the PDF Report Writer server is running on a different computer from the client. To send files to the server for processing you will need to pass them to the TCP/IP port with a special syntax if you are writing your own program (pdfrw_tcp handles this behind the scenes for you). Issue the command -send --binaryname--<filename>--binarybegin--<binary data here>--binaryend--. Note that base64 encoding may be used as well - substitute the text "base64" for "binary" in binaryname, binarybegin, and binaryend. The <filename> must match the name of a file being processed as input. The binary data for that file can come from any file on the client you wish to use to represent that file. For example, here's how you would pass 2 files (using Perl syntax):

```
use IO::Socket;
my $host = 'localhost'; # host server is running on
my $port = '12345'; # port server is running on
my $sock = new IO::Socket::INET (
  PeerAddr => $host,
  PeerPort => $port,
  Reuse => 1,
  Type     => SOCK_STREAM,
  Proto => 'tcp',
    );
print $sock "a.frw -data b.xml t.pdf -send
 --binaryname--a.frw--binarybegin--(a.frw contents)--binaryend--";
print $sock " -send
 --binaryname--b.xml--binarybegin--(b.xml contents)--binaryend--";
print $sock "\nBUILDPDF\n";
```

You may use the option -return to receive the file back via TCP/IP from the PDF Report Writer server. Specify the path and file name you wish to store the output under on the client. The output will not be stored on the server in this case. This allows you to receive the output PDF on the client side that you can then save or process accordingly. Of course, the larger your files the longer it will take to process as your connection speed will play a role in the time it takes to send and receive large PDFs.

The pdfrw_tcp program makes it easier to accomplish the above when transferring files. You may use -send filename.pdf where filename.pdf is the name of the PDF to send. The program will take care of sending the contents of the file in this case. For example:

```
pdfrw_tcp a.frw -data b.xml t.pdf -send a.frw=c:\myfile.frw
  -send b.xml=c:\mydata.xml -return c:\out.pdf
```

In this case, a.frw is the client file c:\myfile.frw and b.xml is c:\mydata.xml. The output as referenced by t.pdf on the server will be sent back to the client and saved as c:\out.pdf. The file t.pdf will not be stored on the server in this case. The data will come back over the same socket connection as binary data if you are writing your own program to communicate with the server. The content length will be passed back first formatted as "Content-Length: n" where n is the number of bytes followed by a blank line and then the data stream. Once the port is closed that is the end of the file.

You may use -sendcache filename.pdf to send the file only the first time you call the server program. The filename.pdf should be the same path and file name of one of your input files. The server will cache the file the next time you need it on future calls to the server. Include the -sendcache option each time you run the program with the file name even though the file will only need to be transmitted once. This can be useful when you have the same background PDF, for example, you wish to reuse many times.

Do not include interactive options such as -open as part of the commands sent to process unless PDF Report Writer server is running locally. Otherwise, the PDF will open on the remote server which is probably not what you intend.

You may want to create a script on the server which will create the necessary data and input file for use by PDF Report Writer. In this case, use the command -exec to provide what script to run. You'll need to include the -allowexec option when you install/start the server to allow -exec. For example:

```
pdfrw_tcp -exec myscript.sh "abc" 123
  -reportwriter #file1 -data #data2
  t.pdf -return c:\out.pdf -clean
```

In this case, the script myscript.sh will be executed on the server and passed the parameters "abc" and 123. The assumption is this script will create two files for input to PDF Report Writer. All the business logic and database connectivity can remain in one location on the server and clients will automatically get new reports whenever the server script is updated. The option -reportwriter is simply a separator between the script command and the options to send to PDF Report Writer. You may need to provide the full path name to the script as well, especially if running the PDF Report Writer server as a Windows service.

The script will need to print or echo the values for #file1 and #data2. Do this by echoing "#file1=<path-file>" during the script execution with each file you need to send back on a separate line. These variables may be named anything you wish but they must start with a # to be converted. For example, have a line that reads "@echo #file1=c:\temp\abc5125.frw" in the script. These will be read after the script finishes and will replace the placeholders #file1 and #data2. You might only have one output file - this example is using two just to show that multiple files may be used. The option -clean instructs PDF Report Writer to clean up or delete the temporary files off of the server.

Here are some sample entries from the log file. In this case, there are 5 simultaneous processes allowed at any one time. The number in parenthesis such as the (1) and (2) below are the pool ids. For example, pool id 1 is used to start a build. While this build is happening, another request comes in to build a PDF. The second request is set to run in thread 2 while thread 1 continues to build.

```
[2010-07-24 16:00:44] Creating pool of 5 entries
[2010-07-24 16:00:44] Accepting commands on port 7075
[2010-07-24 16:01:59] (1) (127.0.0.1) a.frw a2.pdf
[2010-07-24 16:02:00] (2) (127.0.0.1) b.frw b2.pdf
[2010-07-24 16:02:03] (2) (127.0.0.1) finished build
[2010-07-24 16:02:03] (1) (127.0.0.1) finished build
[2010-07-24 16:02:11] (1) (127.0.0.1) -quit
```

The following are the options to use when setting up PDF Report Writer to run as a server. Remember to also include your key name/code combination using -kn and -kc or your software subscription information with -licname, -licpwd, and -licweb.

Option	Description
-server	Used to specify server processing mode. Must be the first option passed.
-host *hostname*	The host name of the computer. The default is localhost.

Client-Server TCP/IP

Option	**Description**
-port *number*	The port number to use. The default is 7075. You may want to setup a descriptive name in etc\services to use instead. For example: pdfrw 7075/tcp Then, use -port pdfrw. By adding this entry in the services files on your clients, you can connect in the same manner by using -port pdfrw. The server will not start if the port is already in use.
-pool *number*	Optional. Pass the number of simultaneous builds to allow at a time. You should start with 5 and increase if you find users are waiting on connections. The log file will show the pool id number used for each build. If you see the maximum number of pool entries being used most of the time then you may want to increase. Keep in mind more processor time will be needed to handle more simultaneous requests so you'll need to balance the two.
-dsnfile	Pass a file containing SQL information such as a name, password, and the database to connect to. Clients can use the connection by specifying the DSN name provided rather than passing a name, password, and database. This allows the server to control the access to any databases without the users knowing the login ids or passwords. You will need to include -allowperl as well as part of the server startup or make sure the clients include that option. See the DSN Layout for details on setting this up.
-dsnpwd *text*	The password that was used to encrypt the DSN file (if encrypted by using -dsnencrypt).
-log *path-file*	Optional. The path and name of a file to log requests to.
-logmax *number*	Optional. The maximum size in bytes for a logfile. Once the file reaches the specified size it is renamed with the current date/time appended to the end and a new log file is started.
-allowexec	Optional. Set this if you want to allow clients to use the -exec option to run scripts.

Client-Server TCP/IP

Option	Description
-allowsafe *text*	Optional. Set this if you want to allow clients to only run certain scripts. Pass in the starting characters for what the script(s) will be named. For example, to only allow people to run scripts that start with the text "rwscript" then use that text with this option. This prevents people from passing in commands such as "rm * -r", for example. In addition, any & or | characters are removed before processing the command. You can separate different starting names with a comma such as "rpt,rwrpt".
-v	Optional. Echoes requests to the screen. Not used when running PDF Report Writer Service.

The client programs pdfrw_tcp and pdfrw_gui_tcp have the same options as PDF Report Writer. There are a few additional options you may use shown in the following table. There is a .NET DLL wrapper (which has the functionality of pdfrw_tcp so you don't need pdfrw_tcp) available on the FyTek website as well as versions for all compiled operating systems at https://www.fytek.com/#/detail/clienttcp. Register the .NET DLL with the regasm program to access as a COM DLL from VBScript or any other Windows program that has support for DLLs. The actual location of the PDF Report Writer server doesn't matter when using the client DLL in Windows. That is, the PDF Report Writer server itself may reside on a Linux box but you can use the client DLLs under Windows to call the server. The DLL object to create in your code is FyTek.ReportWriter. The method to call once all others have been set is "buildReport" to start the build.

Option	Description
-host *host*	The host PDF Report Writer server is running on.
-port *number*	The port number PDF Report Writer server is listening on. The default is 7075. You may want to setup a descriptive name in etc\services to use instead. For example: pdfrw 7075/tcp Then, use -port pdfrw. By adding this entry in the services files on your clients, you can connect in the same manner by using -port pdfrw.

Client-Server TCP/IP

Option	Description
-clopen	Client open. Opens the output PDF file in reader on the local machine. You may use -open if you are running off of the same box PDF Report Writer server is running on.
-exec *script [options]*	Pass in the name of the script the server should execute along with any options. This must be the first option passed when using the executable. To call a script called myscript.sh and pass it options "abc" and 123 you would send '-exec myscript.sh "abc" 123 -reportwriter [options]'. Anything after -reportwriter is what will be passed to PDF Report Writer for processing.
-reportwriter	Use this option after the -exec command to let the program know you are finished with the options for -exec and what follows is for PDF Report Writer to process.
-clprint	Client print. Prints the PDF to the default printer on the local machine. Using -print will print the PDF to the default printer from the machine PDF Report Writer server is running on.
-currdir	Sets the working directory for the server or service to be the current directory. That way your file pathing can be relative to the directory you are currently in and not from where the server or service is running from. This will likely only work if you are running off the same machine the server is running on.
-send *name=path-file*	Used to send files to the machine PDF Report Writer server is running on. Set the name to an input file name and path-file to the path and file name you wish to use for that file. For example, "pdfrw_tcp a.frw t.pdf -return t.pdf -send a.pdf=c:\temp\afile.frw". The file a.frw will be taken as c:\temp\afile.frw and sent to the server for processing. For the DLL, you may call this method multiple times if you have more than one file to send.

Client-Server TCP/IP

Option	Description
-sendcache *path-file*	Used to send a file once to the machine PDF Meld server is running on. Once cached, the server will read from a copy it has kept for itself rather than ask for the file to be transmitted again. Include this option with the same file on each build that you wish to use it. The server cache is cleared once the server program is restarted.
-autosend	Used to send all the input files without using -send or setSend for each one. You only need to set this option once. When the server program looks for a file and this option was used it will send a request back to the client requesting the file. The assumption on the server is none of the files being processed are local files.
-return *filename*	Used to return to your local machine the output PDF from PDF Report Writer. The filename specified should be a local file to save the PDF under.

Client-Server TCP/IP

Option **Description**

-serverstat

Returns a report of the server status. The report contains the following information:

```
Current date time      : 2010-08-01
15:00:00
Server started         : 2010-08-01
12:00:00
Requests received      : 75
Bytes received         : 4090
Bytes sent             : 0
Pool size              : 5
Available pool threads : 5
Highest pool thread use: 2
Requests that waited   : 0
```

"Server started" = the date and time the server was started.
"Requests received" = the total number of requests the server has received to process.
"Bytes received" = the total number of bytes sent into the server for requests.
"Bytes sent" = the total number of bytes in returned PDFs sent back to clients.
"Pool size" = the total number of pool entries the server was started with.
"Available pool threads" = the current number of available threads.
"Highest pool thread use" = the most threads that were in use at any one time.
"Requests that waited" = the total number of requests that have had to wait for an entry in the pool to become available in order to run.

-stopid *id*

Used to cancel a currently running ID. The ID number can be found in the log file with the value ID=n where n is the number to pass to this command. For example, -stopid 5 would cancel the process with ID number 5.

Option	**Description**
-wait	Set this option to cause the non-GUI version of pdfrw_tcp to wait until PDF Meld has finished processing before returning. Normally, when you are not receiving back the resulting PDF, pdfrw_tcp will simply send the request to the TCP/IP port and not wait for PDF Meld to perform its processing. This option causes the program to wait until finished so you know the PDF has been built and you can take some further action with it.

DSN Layout

A DSN file on the server allows you to setup one or more database logins for use by clients. The file contains a name the clients can use to reference the connection by along with information about the database connection such as the user id and password. For example:

```
<DSN NAME="dev" SQLDRIVER="Oracle" USERID="someuser@dev" PASSWORD="abc123">
<DSN NAME="prod" SQLDRIVER="Oracle" USERID="diffuser@prod" PASSWORD="xyz555">
```

This file may be encrypted as an added level of security. This will prevent anyone with access to the file from viewing the contents. Use the option -dsnencrypt with PDF Report Writer to create an encrytped copy. You may then remove the plain text copy and pass the encrypted version to -dsnfile. Be sure to use -dsnpwd when starting PDF Report Writer in server mode so it can decrypt the file.

Clients do not need to be provided with the database details such as a user id or password since it will be contained within this file. The client simply references the connection via the DSN name. For example:

```
<QUERY NAME="myquery" DSN="dev">
```

The DSN file is a tag based file containing the following options.

Parameter	Description
NAME="text"	The DSN name to use. Must be unique in the file. This is the name that the client will need to use the connection on the QUERY tag.
MAXCONN=number	Optional. The number of connections to maintain. Connections are maintained between calls to Report Writer when this option is used so they can be reused, saving time of logging back into the database. If the maximum number of connections are in use then a process needing the connection will have to wait until one becomes available.

Client-Server TCP/IP

Parameter	Description
SQLDRIVER=text	The data source. This is a case-sensitive string. Not necessary when using the DSN option. Available for Windows/Linux only. Valid values are: CSV (or CSVPP) Fixed Tab Pipe Oracle (32-bit only) mysql (or mysqlPP) ODBC XML CSV may give slightly better performance over CSVPP on Windows systems. CSVPP, Fixed, Tab, and Pipe do not support all SQL options.
SQLDB=text	The database schema or driver information. Not necessary when using the DSN option. See the Database Connection section for details.
USERID=text	The user id (if any) for the database connection. For Oracle, you may also specify the password and/or schema in this field. For example, "user/pwd" or "user/pwd@prod".
PASSWORD=text	The password (if any) for the database connection.

PDF Report Writer Service

The PDF Report Writer service is another option when running PDF Report Writer as a server under Windows. See the Client-Server section as the details on the various parameters are covered there. The difference with running as a service is the server program is available to anyone with network access to the server. Plus you don't need to manually start up PDF Report Writer in server mode each time you log in. The service can be set to start whenever the machine is booted so it can be made available without logging in first.

The program pdfrw_srv.exe (or pdfrw_srv64.exe for 64-bit) is the program for the service. You pass in -install as the first option (rather than -server like when running pdfrw.exe) followed by the normal options (such as -pool or -host) that you would use to start in server mode. You'll likely need administrative privileges in order to initially setup the service. Select the "Run as Administrator" option for the DOS box when you go to install.

You'll need to allow TCP traffic on the port if you want to make the service available to other computers. Go into your Windows firewall program and create an entry to allow traffic on that port. You can restrict access by computer and/or user if you like.

Note you still need to pass in a key name/code combination using -kn and -kc or your software subscription information with -licname, -licpwd, and -licweb.

For example:

```
C:\>pdfrw_srv.exe -install auto -pool 5
  -log "c:\logs\rwlog.txt" -host "mymachine"
  -port 7075 -licname "fytek-inc" -licpwd "abc12345"
  -licweb
```

Replace "mymachine" in -host with the actual name of your computer or leave out -host to use the default of localhost. This should start up the service and you will then be ready to start servicing requests. Other options you can use are:

```
c:\>net start PDFRWSrv
```

This will start the service if -install is used without the "auto" option. For example, you can run "pdfrw_srv.exe -install" to simply install the service without starting it.

To stop the service run:

```
c:\>net stop PDFRWSrv
```

This will stop the service.

To remove the service run:

```
C:\>pdfrw_srv.exe -remove -service PDFRWSrv
```

This will remove or un-install the service.

You can test the service by running pdfrw_tcp or pdfrw_gui_tcp. For example:

```
pdfrw_gui_tcp c:\mydir\myinput.frw c:\mydir\myout.pdf
```

Include the -host and/or -port if necessary. If you used the defaults when installing the service and are running on the same machine the service is running on, you can leave out -host and -port. You'll likely want to use the machine name when doing any real testing since the default is localhost for the host and you'll only be able to access from the box the service is running on.

Be sure to fully qualify your file names as the service is not running out of the directory you are running the program pdfrw_tcp from. You can also use the -cwd or -currdir options to change the working directory. The Client-Server section also discusses how you can send files from a remote machine to the server running the service. You can use the -send and/or -return options with pdfrw_tcp in order to send and receive your files to and from the server.

The options for startup as the same as those found in the Client-Server section. The following are additional options for the service.

Option	Description
-username *username*	Optional. The username to run the service as.
-password *password*	Optional. The password for the username. You may leave this option off and, if -username is passed, the program will prompt for the password.

Suggestions

This section lists some suggestions and methods for setting up your report.

Headers/Footers

When setting up a header or footer you may use the Y command to position content correctly. For example, if you want to set up a header with the date on the left, the report title in the middle and the page number on the right, you may do the following:

```
<TEXT FACE=16 SIZE=12 BORDER=1>
<ALIGN VALUE=C><FONT FACE=16 SIZE=12>
Report Title Line 1<BR>
Report Title Line 2<FONT FACE=15 SIZE=10>
<ALIGN VALUE=L>
<Y VALUE=0>Date &date<ALIGN VALUE=R>
<Y VALUE=0>Page &page
</TEXT>
```

Which produces the following:

```
Date Jun 11, 2023          Report Title Line 1          Page 499
                           Report Title Line 2
```

Notice that is not necessary to set the X value during centering or right justification. The program will automatically adjust the X position when the alignment changes. Remember to place content between the <BODY> and </BODY> tags when using headers/footers. If not, your content will overlay the header/footer areas.

Text/Tables

Remember to close out text blocks with the </TEXT> tag if you are mixing text blocks and tables in your report. Use a FONT and SIZE command in the TABLE or TEXT tag when inside of the report BODY. This will ensure the base font for the table or text block is set properly when a page break happens. You may use the BR tag outside of text blocks and tables to give some separation between items. You may also issue a PAGE tag inside of a block of text to force a page break. You may use the TABLEPAGE tag within a TABLE for a page break.

Table Rows

A table row for the outer most table must be able to fit completely on a page. It's important to note that it's the top level table row that must fit. If you have an embedded table inside of a table cell, the entire embedded table is counted as the contents of a cell and must fit on a page. If there isn't enough room for all cells in a given row to print the table will wrap to the next page. Use the CELLSPLIT option on the TABLE or TR tag to specify you want to allow cells to break across pages. Whatever contents will fit at the bottom of a page will print while the rest of the cell contents will start at the top of the following page. Only the text that will fit on a single page is shown if CELLSPLIT is not used on tables with a lot of text in a cell.

Use the CELLXSPACING option for tables that don't have a border. A value of 4 or 5 works good along with CELLSPACING set to 4. This will give some separation to the columns and make it easier to read.

Widgets

Remember to place any INPUT or TEXTAREA tags inside of a TEXT block or table TD or the widget will not be visible. The widget will not show up on the page if don't have it inside one of these. Just treat them as you would any other text item. And don't forget to assign a unique NAME to each widget.

Backgrounds

Use the NEWFORM and USEFORM tags if you find yourself repeating the same information on many pages. This could be text, line drawing commands or images. The advantages are fewer commands passed into the software, the background information is only processed once and the resulting PDF file will be smaller.

Build Options

PDF Report Writer is a CPU intensive program and may take a couple seconds per page to generate the PDF. Partially due to the complexity involved in the PDF structure but mostly to the work that must be done to determine table column widths, perform pagination, reapply all header/footer information and so on. Additionally, PDF Report Writer allows for a variety of text layout commands within a block of text. Because of this, a page can take anywhere from a half-second to three seconds on average to generate depending on processor speed and the complexity of the report. Basically, the faster your CPU, the faster the PDF will generate.

The program will take up as much CPU time as it can get by default. This means the CPU will be at or near 100% usage while the PDF is being built. Here are two things you can do to prevent the system from slowing too much:

- Use the -lowpriority option or setLowPriority method to lower the priority of the program. The program will still use as much of the CPU as it can but other applications will have priority over Report Writer.

- Use the -pause option to allow some idle time for other applications to run. Setting -pause 5 will free up the processor for .001 seconds after every 5 commands. The PDF will build a little slower but the benefit is better CPU utilization.

Keeping the number of commands to a minimum will help decrease the amount of time it takes for the PDF to build. Small things, like putting FONT and BORDER commands in the TABLE tag rather than on each TD tag should be the first thing to look for when trying to decrease build time. Table rows inherit font, border and color characteristics from the TABLE tag. Table cells inherit their font, border and color characteristics from the TH or TR tag. Try to push this information as far up the chain as possible.

Try using the PLAIN option for times when you just need to see the data and are less concerned about formatting. You may apply this option globally or selectively on tables. Another option if you just need to see the data is to use the -csv, -xls or -html command line option and turn off the PDF build. Along with this use the -nopdf option or setNoPDF method to skip the PDF build altogether and use the csv or html output instead. Do not use this option when building RTF output as that output is based on the finished PDF.

Use the CACHE option on your IMG tags for any images that are reused across multiple runs of the program. For example, company logos or other images that remain more or less static. This will cut down on the time needed to convert the image to PDF format each time the program is ran.

The following options may be used if you've optimized your input file and still need a faster build time.

Quick Build

This option will take fewer passes through the file so some features will not be available. You will probably save anywhere from 30-40% over the standard build time using this option. You may not use tables that have the HEIGHT option in the table tag when building with this method. The &getpage function will not work properly when using quick build.

If you use the &totpage variable, the software will leave room for 3 digits since the number is not known while the file is being built. Therefore, the alignment on the text line containing the total number of pages will be slightly off if you have it right or center aligned.

Note there are two quick build methods. The first (-q command line, setQuick method or <QUICKBUILD> tag) builds the PDF in memory. When done, it goes back and puts in information on the total number of pages if that feature was used. The PDF is then written to disk.

The second method (-q2 command line, setQuick2 method or <QUICKBUILD2> tag) works in a similar way except it doesn't go back and plug in the total number of pages where requested. Also, This second method builds the PDF on disk as it goes rather than in memory. Features such as &getpage("ANCHOR") will not work properly with either method however. It's best to use the second quick build method as long as you don't need the total number of pages in the report.

Temp File

Another option, slightly faster than quick build, is to use a temporary file. You will probably save anywhere from 40-50% over the standard build time. This option makes only one pass through the input file so, as with quick build, certain options are not available. You may only use text, tables that do not break vertically, images and bookmarks. You may not use links, added fonts, widgets or embedded PDFs.

If you use the &totpage variable, the software will leave room for 3 digits since the number is not known while the file is being built. Therefore, the alignment on the text line containing the total number of pages will be slightly off if you have it right or center aligned.

Additionally, since there is only one pass of the file, you must provide column width information for each table. The easiest way to do this is to have a hidden TR row just after the table tag. A hidden row is ignored for printing but you can use it to convey the widths of the columns. The widths may be a value based on the current GRID setting or may be a percentage of the grid units in the X direction.

For example, here is how you might set up a table and specify the column widths to use:

```
<TABLE BORDER=1 CELLSPACING=2 CELLPADDING=2>
<TR HIDE></TD>
<TD WIDTH=20%></TD>
<TD WIDTH=45%></TD>
<TD WIDTH=10%></TD>
<TD WIDTH=25%></TD>
</TR>
<TH COLSPAN=2>
... (rest of table) ...
```

In the above example, the table has 4 columns that will span the entire width specified by the GRID command (since the sum of the sizes add up to 100%). You may use less than 100% of the total space in the X direction but don't use more or the table will go outside of the defined area.

Initialization and Table Sizing Sections

You can speed up the build process by use of the INITIALIZE and TABLESIZE tags. These tags identify to Report Writer a small subset of the total report to examine during the build process. The INITIALIZE section marks an area near the top where all document level commands and added fonts are located. The TABLESIZE section marks an area, again near the top, where all table sizing information can be found. See the individual commands for further detailed information on each. Either or both of these options may be used in a given report. You may also combine these options with the quick build option.

Here's an example of a report mocked-up using the two different versions of these commands. This first example shows the opening/closing tags version:

```
<PDF>
<INITIALIZE>
<ADDFONT SRC="myfont.ttf">
<ZOOM VALUE=100>
<QUICKBUILD2>
</INITIALIZE>
<TABLESIZE>
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<PAGE>
<TABLE BORDER=1 NAME=main>
<TR><TD>XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</TD>
<TD>999,999,999.99</TD></TR>
</TABLE>
<TABLE NAME=hdr WIDTH=100%>
<TR><TD WIDTH=20%>Page XXX</TD>
<TD WIDTH=60% ALIGN=C>My Company</TD>
<TD WIDTH=20% ALIGN=R>Date: XX/XX/XX</TD></TR>
</TABLE>
<TABLE NAME=summary BORDER=1>
<TR><TD>This Year</TD>
<TD>99,999,999</TD>
<TD>Last Year</TD>
<TD>99,999,999</TD>
</TR>
<IMG SRC="myimg.gif">
</TABLESIZE>
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<PAGE>
<HEADER MARGIN=2>
<TABLE NAME=hdr WIDTH=100%>
<TR><TD WIDTH=20%>Page &page</TD>
<TD WIDTH=60% ALIGN=C>My Company</TD>
<TD WIDTH=20% ALIGN=R>Date: &date</TD></TR>
```

```
</TABLE>
</HEADER>
<FOOTER MARGIN=2>
<TEXT ALIGN=C>
<IMG SRC="myimg.gif">
</TEXT>
</FOOTER>
<BODY>
<TABLE BORDER=1 NAME=main>
<TR><TD>Data for cell 1</TD>
<TD>100,000.00</TD></TR>
<TR><TD>Data for cell 2</TD>
<TD>5,000.00</TD></TR>
(...rest of the table...)
<TR><TD>Data for cell n</TD>
<TD>15,000.00</TD></TR>
</TABLE>
<BR>
<TABLE NAME=summary BORDER=1>
<TR><TD>This Year</TD>
<TD>6,500,000</TD>
<TD>Last Year</TD>
<TD>5,450,000</TD>
</TR>
</TABLE>
</BODY>
```

Note the table sizing section contains one sample row for each table used in the report. The text used in the sizing section shows the maximum expected value for which the table columns will be sized. If the actual text in the rest of the report is much larger, then the columns may appear sized too small. On the other hand, if the all the text is much smaller, the column sizes may appear too large. When the values are fairly close the sizing issue should be negligible. Remember, the trade off for slight variance in column widths is build speed. Also note that the PAGE and TABLE commands, along with whatever else you might place in the TABLESIZE section, are ignored in the final PDF output.

Another way to use the tags is by using the LINES option on the INITIALIZE and TABLESIZE tags. This allows you to place a single opening tag and scan that many lines for information. This works best when you know everything needed for the report is already near the top. A report with one large table for example. The next example shows the opening/closing version of the tags:

```
<PDF>
<INITIALIZE LINES=50>
<ADDFONT SRC="myfont.ttf">
<ZOOM VALUE=100>
<TABLESIZE LINES=200>
```

```
<GRID ML=.5 MR=.5 MT=.5 MB=.5 XUNITS=80 YUNITS=80>
<PAGE>
<HEADER MARGIN=2>
<TABLE NAME=hdr WIDTH=100%>
<TR><TD WIDTH=20%>Page &page</TD>
<TD WIDTH=60% ALIGN=C>My Company</TD>
<TD WIDTH=20% ALIGN=R>Date: &date</TD></TR>
</TABLE>
</HEADER>
<FOOTER MARGIN=2>
<TEXT ALIGN=C>
<IMG SRC="myimg.gif">
</TEXT>
</FOOTER>
<BODY>
<TABLE BORDER=1 NAME=main>
<TR><TD>Data for cell 1</TD>
<TD>100,000.00</TD></TR>
<TR><TD>Data for cell 2</TD>
<TD>5,000.00</TD></TR>
(...rest of the table...)
<TR><TD>Data for cell n</TD>
<TD>15,000.00</TD></TR>
</TABLE>
</BODY>
```

In the above example, all the information necessary for the initialization piece is located within the first 100 lines. This includes the added font and zoom setting. The table size uses 200 lines to gather enough sample rows to size the table. The assumption is all rows have about the same amount of text so looking at the first 50 rows or so will give a close enough estimate to size the columns by.

Sizing Files

The -insize/outsize options can be used as an alternative to the TABLESIZE tag. These options are used to create a sizing file that you can use for building similar PDFs in the future. It can also be easily used in situations where you have tables near the end of the input file that would difficult to use between the opening and closing TABLESIZE tags.

This works best when you are creating many PDFs all set up the same as far as the tables go but with different values for the cells and/or different number of table rows. For example, a packing slip or customer invoice. You create the sizing file by creating a sample PDF along with the -outsize option or setOutSize method.

```
pdfrw sample.frw -outsize sample.siz
```

For all future builds, use the sizing file to skip the table sizing routines. The PDF will be built faster, especially when using the INITIALIZE tag in the input PDF.

```
pdfrw sample2.frw -insize sample.siz
```

Remember the layout of the PDF must match the sizing file. That is, the same table structure only different data for the cells. The number of rows may be different but the columns must match or you will either not get a valid PDF or the table columns will not be sized correctly. Note this feature has similar limitations to the quickbuild methods. Variables such as &getpage("ANCHOR") and &totpage will not work properly since they are determined by computations that are bypassed for the option.

Error Checking

You can check your PDF commands for errors during your development stage. When you use this option, any errors found are written to the error log file along with the line number from the input file that caused the error. The error log file will not be created if no errors are found. If errors are found, the PDF file will not be created.

You turn on error checking in the executable or Perl version with the -e parameter. For example: "pdfrw filein.frw fileout.pdf -e errlog.dat" will place the error information in the errlog.dat file.

The following table lists possible errors and their cause.

Error	Description
FONT xxxx not found	The font file specified in an ADDFONT tag could not be found. Check the path on the FILE parameter and the spelling.
IMAGE xxxx not found	The image file specified in an IMG tag could not be found. Check the path on the SRC parameter and the spelling.
TEXT tag not closed	An opening TEXT tag was found but no closing /TEXT tag was found.
BODY tag not closed	An opening BODY tag was found but no closing /BODY tag was found.
BODY tag encountered while still in HEADER/FOOTER	A BODY tag was found before either the HEADER or FOOTER was closed. You must close out a HEADER or FOOTER before issuing the BODY command.
HEADER/FOOTER tag encountered while in a TABLE	A HEADER or FOOTER tag was found while in a TABLE. Close out the TABLE before issuing the HEADER/FOOTER command.
BODY tag encountered while in a TABLE	A BODY tag was found while in a TABLE. Close out the TABLE before issuing the BODY command.

Error Checking

Error	Description
HEADER/FOOTER tag encountered while still in BODY	A HEADER or FOOTER tag was issued while inside of the BODY. Close out the BODY tag first. If you are using HEADERS/FOOTERS they should always come after the PAGE command and before the BODY command.
TD tag with no active TABLE	A TD tag was found but no TABLE command had been issued. Open the TABLE first, followed by a TH or TR then the TD command.
TD tag with no TH, TR or /TD tag prior	A TD tag was found but was not preceeded by a TH, TR or /TD command. A TD command should only follow a TH, TR or /TD command.
/TH tag with no /TD tag prior	A closing /TH tag was found but was not preceeded by a /TD command.
/TR tag with no /TD tag prior	A closing /TR tag was found but was not preceeded by a /TD command.

Index of Commands