SimplePDF Composer ver 2.02
Help System Improvements (proposed)

This file includes proposals for improving the on-line help for this product. When you use the product for testing, pretend that the workspace has been changed as shown on the next page.

I have proposed changes to some of the icons to conform to those used in popular applications.

You can use the help index that starts on page 7 of this document.

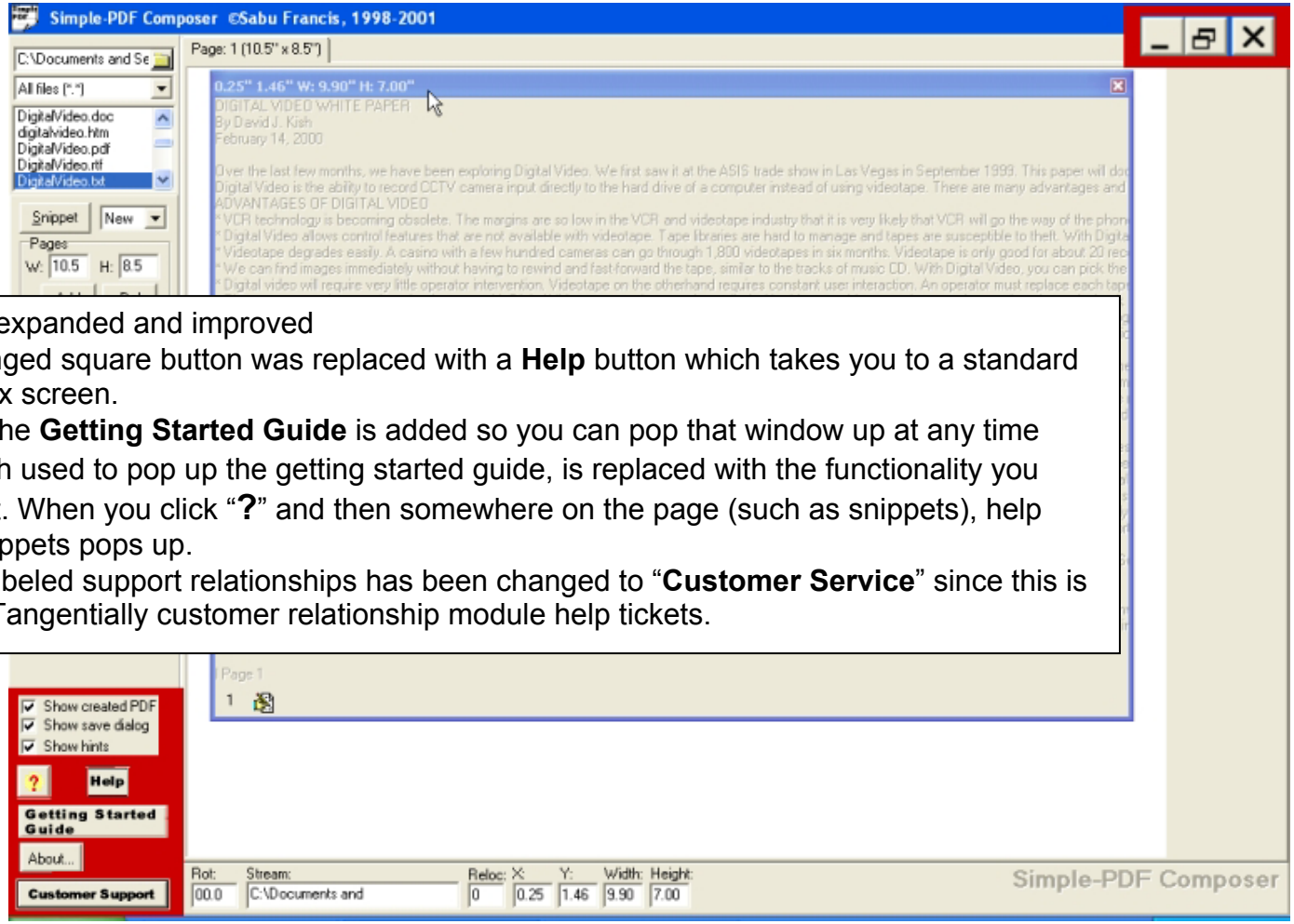
The help index was created by searching through all existing documentation found in the simplePDF directory. Some of those files were part of the information shown during installation, the manual you can print from within it, and other files that are not referenced by the application. I made little attempt at organizing, and made a few corrections of blatant spelling errors. I chose what is included because it is representative of existing text the vendor could have used to create an on-line help system. However, I did not attempt to organize or rewrite any of the text.

If you participated in the first phase, the information in the help index may explain some of the things you found confusing.

Note: None of these changes have actually been incorporated in the application.

SimplePDF Composer ver 2.02
Workspace Improvements (proposed)

Improvements to the workspace are highlighted in red. They include removing the unusual floppy disc icon at the top right corner, as well as expanding and redefining the help section.

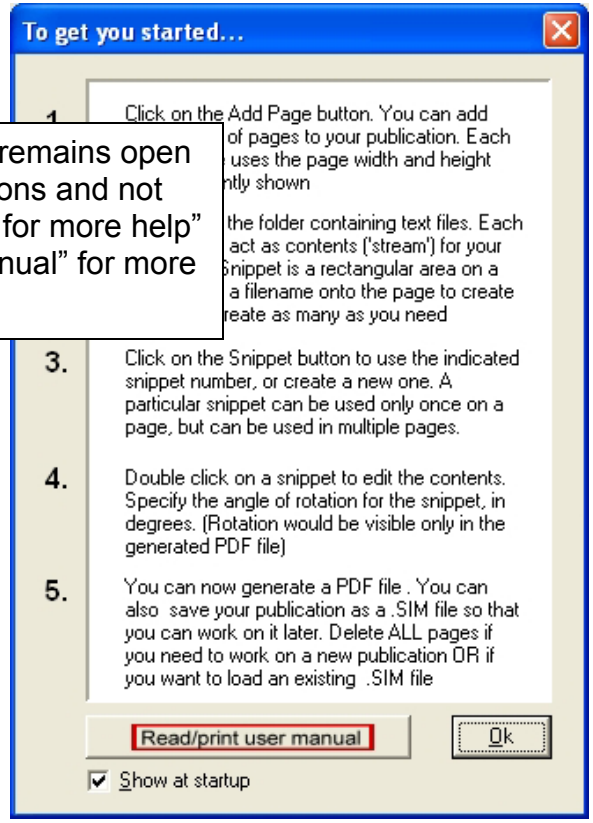


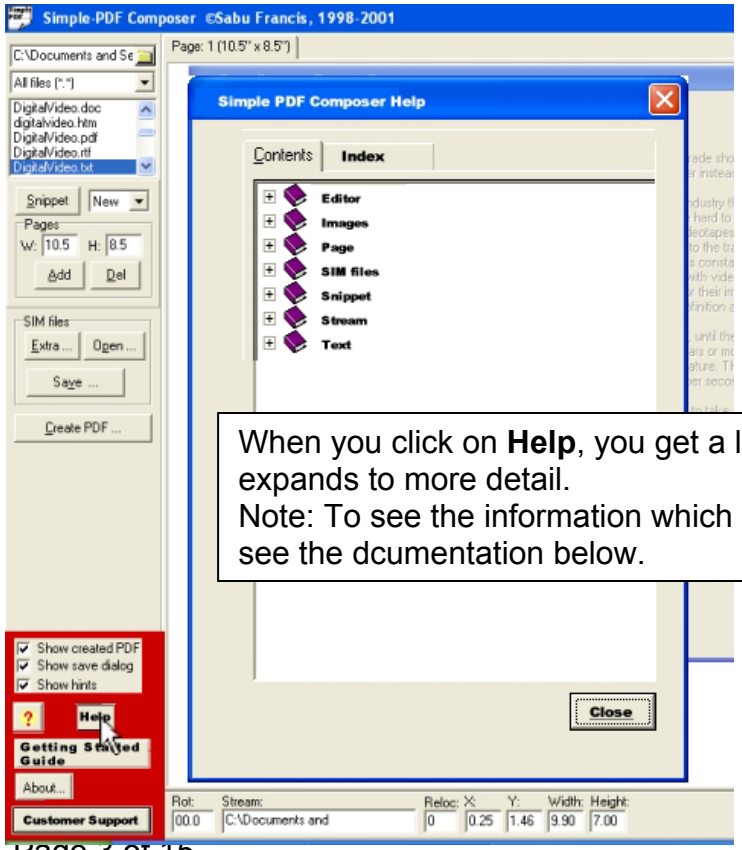
The help section is expanded and improved

- The green tinged square button was replaced with a **Help** button which takes you to a standard content, index screen.
- A button for the **Getting Started Guide** is added so you can pop that window up at any time
- The "**?**" which used to pop up the getting started guide, is replaced with the functionality you would expect. When you click "**?**" and then somewhere on the page (such as snippets), help related to snippets pops up.
- The button labeled support relationships has been changed to "**Customer Service**" since this is a link to the Tangentially customer relationship module help tickets.

SimplePDF Composer ver 2.02
Help System Improvements (proposed)

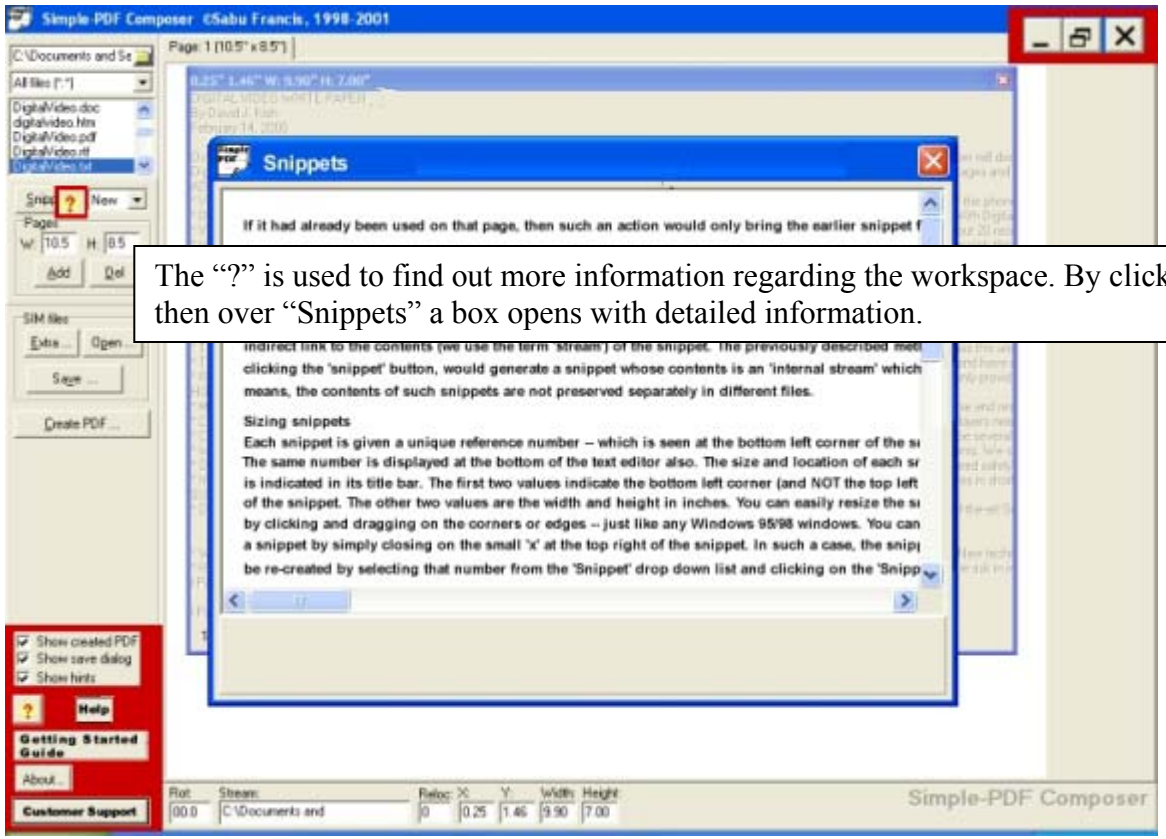
The getting started guide has been changed so that it remains open until you close it. This allows you to follow the instructions and not lose your place in the guide. The “Read composer.pdf for more help” button label has been changed to “Read/print user manual” for more clarity.





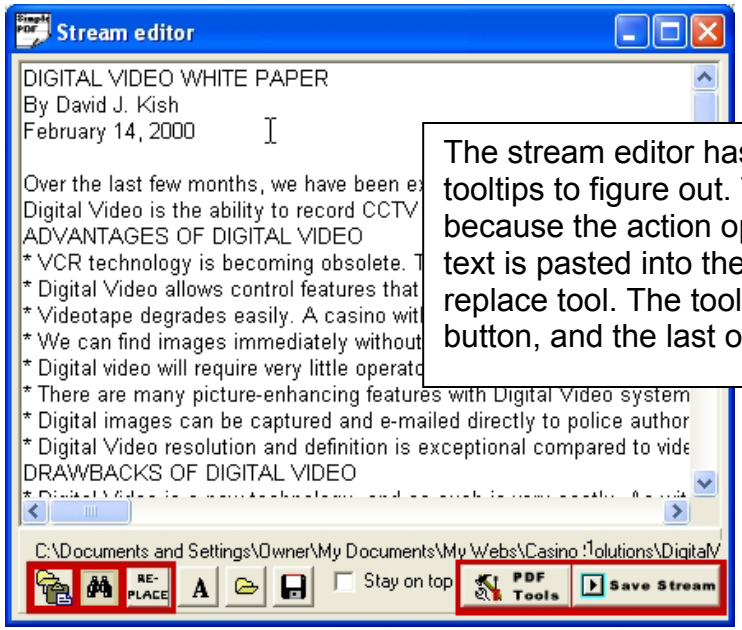
When you click on **Help**, you get a listing of basic topics that expands to more detail.
Note: To see the information which should be included here, see the dcumentation below.

SimplePDF Composer ver 2.02
Help System Improvements (proposed)



The "?" is used to find out more information regarding the workspace. By clicking on the "?" and then over "Snippets" a box opens with detailed information.

SimplePDF Composer ver 2.02
Stream Editor and Error Message Improvements (proposed)

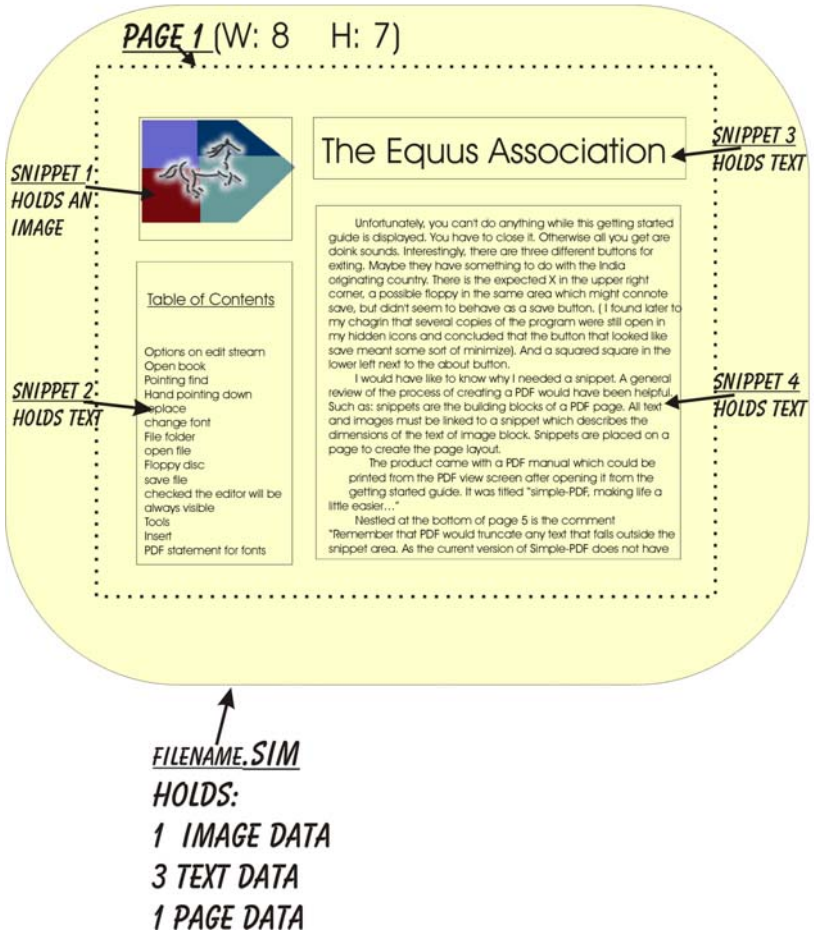


The stream editor has been modified by replacing unusual icons with ones you don't need tooltips to figure out. The first one on the bottom left is a combination open file and paste because the action opens a file where you highlight text and when you close the file, the text is pasted into the editor. The second one is a find text tool, the third one is a find and replace tool. The tools button seems to make more sense to describe it as a PDF Tools button, and the last one, though unusual, benefits from the description "Save Stream".

Add more information to the error messages. For instance, if you try to open an illegal file name, the error message is been expanded to show what types of files are acceptable.

SimplePDF Composer ver 2.02
Overview Diagram in Help (proposed)

Anatomy of a SIM file



SimplePDF Composer ver 2.02
Help Index Entries (proposed)

IMAGES

Snippets can be text or images

(From version 1.8 onwards, you can even place JPEG images in your publications. File extensions: .jpg, .jpeg, and .jif are recognized)

JPEG image files can be inserted into a Simple-PDF publication

you can use JPEG image files too (those files; obviously cannot be edited using any text editor!)

TEXT

You can edit the individual snippets using a simple in-built text editor. You can also layout the snippets on individual pages. The editor is bare but it has the necessary features required of a text editor. We would love to develop a full fledged WYSIWYG tool for you -- but for that we need money

Remember that PDF would truncate any text that falls outside the snippet area. As the current version of Simple-PDF does not have text metrics capability, it may inadvertently crop your text.

Create multiple page PDF files from plain ASCII text files.

Why would you want to use external stream files, you may wonder? It can be very useful when you are creating several PDF files all of which share some common text files.

The editor also has a useful utility to insert PDF statements within your text. We've given the most popular ones. For annotations and links, you would have to construct special statements directly in your snippet stream as explained in simplepd.pdfs

If you want to shuffle pages around, you would have to edit the .SIM file separately in a text editor before using it in The Composer.

At the simplest level, a stream can be any ole ASCII text file. When you want more sophistication, you can selectively put native PDF marking operators into the text file. As you get into more and more sophistication, you would embed more and more PDF marking operators into your stream.

It may be advantageous to churn out one long text stream file at first. There is a chance that all of that would not fit into one page. Once you are ready to get into the publication design of your project, you could then divide that large stream file into parts and create separate stream files for each part; and use them separately on consecutive pages in your publication.

After you decide on the snippet size, location and page size of your PDF publication, create one stream text file for each of the snippet on one page. Then check out if the margins are not being exceeded in any direction. If the stream flows out of the margins, Acrobat reader would simply crop it out, so you must know how much to type on each line in your stream file.

It would help if you can use a font in your text editor that matches the one that would be finally used in your publication. For example; I use MS-SansSerif in my editor, to somewhat match up to Helvetica that is used by Acrobat Reader

You SHOULD NOT use automatic word-wrap feature in your text editor for your streams, if such a feature is available. Instead, you must create individual lines in the text file by consciously pressing the ENTER key at the end of each line. Simple-PDF does not have the ability to break lines as per the margins available. You would have to set the margins by trial and error.

If you have a very small snippet, then it may not be worth creating a separate text file for its stream. You can simply write the stream directly into the .SIM file instead of the stream filename.

Expected changes in future versions
Automatic wordwrapping and automatic formation of repeating snippets in newer pages, as per font used

SNIPPET

Simple-PDF uses the concept of snippets and pages to create any complicated PDF book.

Every PDF publication created by SIMPLE-PDF would consist of 3 kinds of elements: Pages, Snippets and Streams

Insert vector graphics inside snippets

Insert bitmap graphics (JPEG format) as snippets.

You can edit the individual snippets using a simple in-built text editor. You can also layout the snippets on individual pages

Each snippet appears on the page as a gray, empty dialog box with the Snippet number in the bottom left corner.

Double click on it to edit its contents. Resize the snippet by clicking and dragging the mouse at the corners of the snippet 'dialogs' - just like the way you would do it with other regular Windows. Remove unwanted snippets by clicking its close box.

When you want to add a new snippet (i.e. a rectangular area on the screen), make sure that the drop down list next to the 'Snippet' button is pointing to 'New'. If it was showing some other snippet number, then The Composer would attempt to use that same snippet on the currently active page

If it had already been used on that page, then such an action would only bring the earlier snippet forward of the others -- it will not allow you to use the same snippet on the same page twice. Using the file and folder controls at the top of the controls area of The Composer, you could also drag a filename onto the white region. When you leave the mouse, a snippet would be created for you at the location where you had lifted the mouse button. When you create a snippet in this fashion, The Composer would keep an indirect link to the contents (we use the term 'stream') of the snippet. The previously described method of clicking the 'snippet' button, would generate a snippet whose contents is an 'internal stream' which means, the contents of such snippets are not preserved separately in different files.

Sizing snippets

Each snippet is given a unique reference number -- which is seen at the bottom left corner of the snippet. The same number is displayed at the bottom of the text editor also. The size and location of each snippet is indicated in its title bar. The first two values indicate the bottom left corner (and NOT the top left corner) of the snippet. The other two values are the width and height in inches. You can easily resize the snippets by clicking and dragging on the corners or edges -- just like any Windows 95/98 windows. You can delete a snippet by simply closing on the small 'x' at the top right of the snippet. In such a case, the snippet can be re-created by selecting that number from the 'Snippet' drop down list and clicking on the 'Snippet' button adjacent to it. The contents (i.e. stream) of such deleted snippets, are not lost if you now decide to bring the Snippet back to life after deleting. However, any saving action would flush out all the deleted snippets completely. If the stream was preserved as an external file; then that would be left untouched even if the snippet using it was deleted. At the bottom of the main screen, the salient details of the currently active snippet are also shown.

From left to right, they are: the rotation value, the stream contents (or the filename, if it is an external stream) and once again, the bottom left corner (X,Y) as well as the width and height of the snippet. Double-clicking on any of those sizing information can also change the size and location of the active snippet. If you change the rotation to any angle within, a colored strip would appear at the bottom of the snippet indicating that the indicated snippet would be finally rotated at the specified angle. Please note: Rotated snippets would not appear in the same rectangular region as shown within The Composer. The size and position of the snippet BEFORE rotation is what is shown. Please also note that you would have to click inside a particular snippet at least once, for the program to register the new values. Shifting from one page to the other will NOT make the program automatically shift the snippet information it is showing: If you now double click at the 'Stream:' edit box at the bottom of the screen, you may still be accessing the earlier snippet and not the new one that is visible on the newly selected page. In short, when in doubt, click on the snippet once before editing

Editing snippets

To edit the contents of any snippet, whether internal or external; simply double-click on the snippet itself. The Composer meticulously preserves the last editing caret location, so even if you shift from one snippet stream to another, the editing caret will be faithfully waiting for you at the same location when you return. You can also double click on the 'Stream' part at the bottom of the screen to edit the contents of the snippet -- even if it was an external file. It is recommended that you resize the width of the editor to suit the line length you may want. You may have to produce some trial PDF files before you get comfortable. Remember that PDF would truncate any text that falls outside the snippet area. As the current version of Simple-PDF does not have text metrics capability, it may inadvertently crop your text.

Multiple pages from one snippet

From version 1.5, onwards, you can create multiple-pages from one snippet itself! Just write down: <page> where you need the page break to occur. At that point, the program would generate a new page; using the same snippet size parameters. (Note: There is NO space after the < and before the > characters, and such a command should be placed at the beginning of a new line, and there should NOT be any other characters on

that line, even blank spaces! If you are wondering how I managed to write down the <page> above; this is how it happened: I added a few space characters just after the <page> Thus that particular line was NOT translated into a page-break.) Other snippets that you may have designed on that page would not get replicated when the multiple pages are formed. This feature can be very useful when you are writing a book with several chapters, and you do not know the number of pages in each chapter. Simply create a separate page for each chapter; and in the main snippet on the 'chapter-page' use the <page> feature to create multiple pages! You can see this feature used in the composer.sim file. Go to page 5 of the SIM file and examine the contents of the main snippet. You would notice that this particular feature enables the program to break it into two separate pages. Thus the generated PDF file would have one extra page than what was seen in the .SIM file When you use this feature; remember the following points:
a) Many times, we would start the page with some pdf(...) font commands. These would have to be repeated after each such page-breaks, else you would notice that the font reverts back to the default 10 point helvetica.
b) It is not advised to use the page break command for more than one snippet in any given designed page in the .SIM file (By the term 'designed page' I mean the page, as it
appears in the .SIM file. When the .PDF file is eventually generated, it could have many more pages because of this 'page break' feature.
c) When a page break is discovered within a snippet; the program would break into a new page using the same snippet location and size. Other snippets are not used if their 'Reloc' parameter is 0 (zero). You can change this behaviour by using a 'Reloc' value of 1 for a snippet that needs to be repeated for every page automatically created using such page-breaks
d) You can insert page numbers for such generated pages by using the $ sign followed by the small case letter 'p'. Thus; this is page 2. (See the .SIM file to understand what I wrote here) If you notice; such a page number starts at 1 for every such snippet containing pagebreaks. Load novel.sim to see how you could use the page-break feature for writing a novel...

When you do any file saving, the program would flush out all unused snippets and all the snippets would get renumbered, if necessary.

A 'Snippet' is a rectangular part of a page. The same snippet can be used on many different pages, and if so required it can be relocated in different locations on those pages. However, a snippet can be placed only once on any individual page. You can have as many snippets as you need on a page.

You must have at least one snippet instruction in your .SIM file and you can have as many as you require.

The 'indexNumber' is a number used for indicating each snippet. You must use contiguous numbers for your snippets starting from 1. If you do not, you would get a corrupted PDF file. The index number of each snippet should also be unique.

The same snippet may be used in many pages (for example; as title blocks, etc.)

Multiple pages can be created from each snippet. Just insert <page> on a newline all by itself (no spaces before or after that <page>). Acrobat Reader is called automatically for previewing in Simple-PDF Composer

Repeating snippets
(Create any block and make it repeat on all your pages, without you having to redo any work. In fact the 'snippets' feature is central to the working of SIMPLE-PDF. The Vertical logo on the side of the pages is an example of a repeating snippet) Snippets can be text or images (From version 1.8 onwards, you can even place JPEG images in your publications. File extensions: .jpg, .jpeg, and .jif are recognized)

STREAM

Every PDF publication created by SIMPLE-PDF would consist of 3 kinds of elements: Pages, Snippets and Streams

But the third concept, a 'stream' talks about the actual contents that go into the PDF publication. Just like a 'stream' in the real world, it flows! But this time it brings in text and graphics into a snippet.

A stream, as stated before, is some data that flows into a snippet. At the simplest level, a stream can be any ole ASCII text file. When you want more sophistication, you can selectively put native PDF marking operators into the text file. As you get into more and more sophistication, you would embed more and more PDF marking operators into your stream.

The Composer would keep an indirect link to the contents (we use the term 'stream') of the snippet.

clicking the 'snippet' button, would generate a snippet whose contents is an 'internal stream' which means, the contents of such snippets are not preserved separately in different files. The data of 'internal streams' are therefore stored within the .SIM file.

Why would you want to use external stream files, you may wonder? It can be very useful when you are creating several PDF files all of which share some common text files.

You can also double click on the 'Stream' part at the bottom of the screen to edit the contents of the snippet -- even if it was an external file

For annotations and links, you would have to construct special statements directly in your snippet stream as explained in simplepd.pdf

Annotations and links are to be created using the native commands in individual streams

Normally, when you start work on your publication, you should first be concentrating on the various streams that go into the snippets of your PDF publication. These stream files can be created in any ascii text editor that handles multiple files. From ver 1.3 onwards, Simple-PDF also contains its own interactive program, Simple-PDF Composer; that will do nicely.

Apart from the .SIM file, you would usually need a set of ascii text files for each of the streams used in your publication. The names of those stream files would be written down in the appropriate locations within the instructions found in the .SIM file

It may be advantageous to churn out one long text stream file at first. There is a chance that all of that would not fit into one page. Once you are ready to get into the publication design of your project, you could then divide that large stream file into parts and create separate stream files for each part; and use them separately on consecutive pages in your publication.

You SHOULD NOT use automatic word-wrap feature in your text editor for your streams, if such a feature is available. Instead, you must create individual lines in the text file by consciously pressing the ENTER key at the end of each line. Simple-PDF does not have the ability to break lines as per the margins available. You would have to set the margins by trial and error.

SIMPLE-PDF produces unoptimised PDF files. It does not encrypt the streams, nor does it compress them.

Reading of stream files have now been optimized. Now they are read only once; and therefore you save time! Annotations and links can now be given in files that are asked during runtime.

A nifty JavaScript utility that works in Netscape or IE (both 4 and above) is included for creating/editing .SIM files. Double click on makesim.html and the unique Javascript application will run.

SIM files

The command line utility is also included with Simple-PDF Composer, and as both the programs use the same .SIM file, you can switch from one to the other to create and manage PDF publications. Simple-PDF stores a publication using the .SIM file format. Before this version, the only way to create a .SIM file is to read the documentation for Simple-PDF, understand the sample .SIM file that was provided and create such files yourself. From this version onwards, The Composer can be used to create the .SIM file for you. Thus the Composer is a companion tool for the Simple-PDF command line utility.

If you need convenience, then use The Composer. But if you already have a .SIM file ready, which you want to reuse then use the command line utility. The Composer is quite intuitive to use: Here are the steps to create a publication:
a) Create a set of pages using the 'Add Page' button.
b) Assemble snippets on each page. Each snippet appears on the page as a gray, empty dialog box with the Snippet number in the bottom left corner.
c) Double click on it to edit its contents. Resize the snippet by clicking and dragging the mouse at the corners of the snippet 'dialogs' - just like the way you would do it with other regular Windows. Remove unwanted snippets by clicking its close box.
d) Once you are ready, click on 'Create PDF...' to generate the PDF file.
Before quitting, click on the 'Save SIM file...' button so that the publication can be saved as a SIM file, to be reused later.

It is not advised to use the page break command for more than one snippet in any given designed page in the .SIM file (By the term 'designed page' I mean the page, as it appears in the .SIM file. When the .PDF file is eventually generated, it could have many more pages because of this 'page break' feature.

Before you quit the program, it is better that you save the .SIM file so that you can work on the same publication some other time. Having generated a PDF file will not give you the freedom to work on the publication again: You would require the .SIM file from which the PDF was generated If you need a tutorial, you could load composer.sim into The Composer. That is the SIM file that was used to create this PDF file. If you need to learn the internal details of the .SIM files, please read 'Simplepd.pdf' That document contains complete details on the .SIM file format. There are a few things that Simple-PDF Composer is currently not capable of directly handling, and these features would be incorporated in future versions. These include:
a) Using a GUI to handle Table of Contents (Bookmarks). Currently all advanced commands
can be given within The Composer, but you would have use the native .SIM instructions
b) If you want to shuffle pages around, you would have to edit the .SIM file separately in a text editor before using it in The Composer.

c) Annotations and links are to be created using the native commands in individual streams
d) Simple-PDF is also capable of handling vector graphics. This requires knowledge of some parts of the PDF file format. We are in the process of creating Simple-PDF Publisher which will have all these features built in.

Apart from the .SIM file, you would usually need a set of ascii text files for each of the streams used in your publication. The names of those stream files would be written down in the appropriate locations within the instructions found in the .SIM file

The SIM instruction file-This file contains the actual instructions for forming the structure of the PDF publication. Each instruction is to be given in a separate line, exactly as indicated below. Even if the instruction becomes very long, each instruction MUST NOT SPAN more than one line.

You must have at least one snippet instruction in your .SIM file and you can have as many as you require.

Dimensions and sizes in Simple-PDF instructions (both in the .SIM file and the stream files) are given in inches (excepting for font size which is always in 'points'). But pure PDF marking operators use a units that scale this way: 72 units = 1 inch. You must appropriately calculate the actual values when you use pure PDF operators in the pdf(...) instruction embedded in your stream file. Also, the pdf(...) instruction should end in a space: For e.g. Notice a space after rg: pdf("0.3 0.1 0.8 rg ") If not, each pdf(...) would nudge into the next one.

In SIMPLE-PDF you would have to first print out the page containing the links, physically calculate the rectangles that you would need by measuring it off the printed page and then revise the stream files to incorporate the link rectangles. Actually it is no fault of SIMPLE-PDF, Adobe has designed PDF to be that way.

SIMPLE-PDF as presented in this fashion (i.e. using SIM files and stream files) could be argued as replacing one complex format (PDF) with another (SIM). To be productive with SIMPLE-PDF it would have been best to have a front-end that would do the physical positioning of the snippets, etc. (Simple-PDF Composer is such a front end)

a complete publication can be stored as one .SIM file. This was not possible in earlier versions, where large streams had to be stored as separate files. Simple-PDF now has a companion utility called Simple-PDF composer that can be used to create .SIM files interactively
ipt application will run.

Both the command line utility as well as The Composer, uses the same .SIM file format for creating PDF files; and they can be used interchangeably

PAGE

Simple-PDF uses the concept of snippets and pages to create any complicated PDF book.

Create multiple page PDF files from plain ASCII text files.

Arrange rectangular regions called snippets on individual pages

Create a set of pages using the 'Add Page' button.

About pages

When you create a page, The Composer would show you a large white region reflecting the size of the page that was last set. If you need another size for the page, you can set the size using the controls just above the 'Add Page' button. Or you can change the page size by editing the width and height and double clicking on either of the edit boxes. Each page maintains its own page size separately. Clicking 'Del' to delete a page would prompt you for confirmation. If you had pressed the 'Shift' key, then a page would be deleted without any prompt. If you press 'Ctrl' when you click on Del, then all pages would get deleted, after asking you for a confirmation

Multiple pages from one snippet

From version 1.5, onwards, you can create multiple-pages from one snippet itself! Just write down: <page> where you need the page break to occur. At that point, the program would generate a new page; using the same snippet size parameters. (Note: There is NO space after the < and before the > characters, and such a command should be placed at the beginning of a new line, and there should NOT be any other characters on that line, even blank spaces! If you are wondering how I managed to write down the <page> above; this is how it happened: I added a few space characters just after the <page> Thus that particular line was NOT translated into a page-break.) Other snippets that you may have designed on that page would not get replicated when the multiple pages are formed. This feature can be very useful when you are writing a book with several chapters, and you do not know the number of pages in each chapter. Simply create a separate page for each chapter; and in the main snippet on the 'chapter-page' use the <page> feature to create multiple pages! You can see this feature used in the composer.sim file. Go to page 5 of the SIM file and examine the contents of the main snippet. You would notice that this particular feature enables the program to break it into two separate pages. Thus the generated PDF file would have one extra page than what was seen in the .SIM file When you use this feature; remember the following points:

a) Many times, we would start the page with some pdf(...) font commands. These would have to be repeated after each such page-breaks, else you would notice that the font reverts back to the default 10 point helvetica.

b) It is not advised to use the page break command for more than one snippet in any given designed page in the .SIM file (By the term 'designed page' I mean the page, as it appears in the .SIM file. When the .PDF file is eventually generated, it could have many more pages because of this 'page break' feature.

c) When a page break is discovered within a snippet; the program would break into a new page using the same snippet location and size. Other snippets are not used if their 'Reloc' parameter is 0 (zero). You can change this behavior by using a 'Reloc' value of 1 for a snippet that needs to be repeated for every page automatically created using such page-breaks

d) You can insert page numbers for such generated pages by using the $ sign followed by the small case letter 'p'. Thus; this is page 2. (See the .SIM file to understand what I wrote here) If you notice; such a page number starts at 1 for every such snippet containing pagebreaks. Load novel.sim to see how you could use the page-break feature for writing a novel...

The editor also has a useful utility to insert PDF statements within your text. We've given the most popular ones. For annotations and links, you would have to construct special statements directly in your snippet stream as explained in simplepd.pdf

If you want to shuffle pages around, you would have to edit the .SIM file separately in a text editor before using it in The Composer.

From ver 1.5 onwards, you can create multiple pages. Simply write <page> on a new-line all by itself wherever you need a page break. If you use $ p (the letter p shall immediately follow the $, without any space) then that is substituted by the page number created for multiple pages. Refer novel.sim for an example

It may be advantageous to churn out one long text stream file at first. There is a chance that all of that would not fit into one page. Once you are ready to get into the publication design of your project, you could then divide that large stream file into parts and create separate stream files for each part; and use them separately on consecutive pages in your publication.

The PDF file format is quite vast and covers all aspects of page creation. However the PDF file format is daunting for common tasks and it is almost next to impossible to convert simple text files (such as the one you are now reading) into a PDF publication simply by hand.

EDITOR

The editor is bare but it has the necesary features required of a text editor. We would love to develop a full fledged WYSIWYG tool for you -- but for that we need money.

It is recommended that you resize the width of the editor to suit the line length you may want.

The editor also has a useful utility to insert PDF statements within your text. We've given the most popular ones. For annotations and links, you would have to construct special statements directly in your snippet stream as explained in simplepd.pdf

If you want to shuffle pages around, you would have to edit the .SIM file separately in a text editor before using it in The Composer.

stream files can be created in any ascii text editor that handles multiple files. From ver 1.3 onwards, Simple-PDF also contains its own interactive program, Simple-PDF Composer; that will do nicely

It would help if you can use a font in your text editor that matches the one that would be finally used in your publication. For example; I use MS-SansSerif in my editor, to somewhat match upto Helvetica that is used by Acrobat Reader.

You SHOULD NOT use automatic word-wrap feature in your text editor for your streams, if such a feature is available. Instead, you must create individual lines in the text file by consciously pressing the ENTER key at the end of each line. Simple-PDF does not have the ability to break lines as per the margins available. You would have to set the margins by trial and error.

You can edit the individual snippets using a simple in-built text editor. You can also layout the snippets on individual pages. The editor is bare but it has the necessary features required of a text editor. We would love to develop a full fledged WYSIWYG tool for you -- but for that we need money.