The following conventions are used in all documentation for Power Page. 'Value' can either be a literal value or a cell referenced value. This holds true even when a descriptive prefix or suffix is added to it such as 'max_value' or 'value_1.' 'Cell_ref' must be a cell reference, literal values aren't allowed. 'Filename' refers to a literal file name although, in most cases, a cell reference containing a file name would work just as well. 'Logicop' refers to any of the normal logical operations (<, >, =, <=, >=, or <>).

Power Page expects certain conventions in terms of handling volumes in HFS. All text files associated with a given power page document are expected to be in the same volume and folder. That volume is assigned when a power page document is opened up. What this means for the user is, when creating a new power page document, all text files should be on the default volume from which the program was loaded. Once the document is saved, however, the text files must be moved to the same volume and folder as the Power Page document. The one exception to this rule is 'pict' files. These files are created when a Power Page document is saved that contains Pict cells. These files will always be located on the default volume where the application lies.

Calculated Cells:

Calculated cells work in one of two ways. If the formula begins with an equals (=) sign, it's handed off to the formula engine to calculate the value when a recalc is done. If there is no equals sign, the text is transfered in whole to the result.

Calculations are done normally following all the conventions for parentheses (up to 16 levels of parentheses). The standard math functions (+ - / * ^) are supported as well as several special functions. Some special functions work as normal math functions, while others work only on data cells. In either case, the special function must be preceeded by parentheses if they aren't the first argument in the formula. For example, "ABS(10/2.5)" is valid. If that calculation wasn't at the beginning it would have to be: "200+(ABS(10/2.5))."

Normal special functions:

ABS(value)	absolute value
INT(value)	truncated integer of value
ATN(value)	arctangent of value
COS(value)	cosine of value
SIN(value)	sine of value
TAN(value)	tangent of value

Data cell special functions:

MIN(cell_ref)	minimum value in list
MAX(cell_ref)	maximum value in list
NDX(cell_ref,value)	retrieves item number pointed to by value in list
CNT(cell_ref)	count of items in list
SUM(cell_ref)	sum of list
AVG(cell_ref)	average of list

Text Cells

The format for text cells is: filename[, find_value, replace_value, find_value, ...replace_value]. Up to 10 find, replaces can be in a formula. The find_value is a single character while the replace_value can be up to 40 characters. It's not necessary to have an entry for multiple find/replaces if they are the same. In other words, if you had 20 instances of the character '*' in the text that you wanted to be replaced with 'now', you would only need '*, now' included in the formula one time to take care of all find/replaces.

The way the user should handle the find/replace function is, type out the document using any single special character they want for a given replacement. Save that document as a text file, then load it into Power Page using the formula for find/replace.

The maximum length of any text cell is 2040 characters, including find/replaces. Text cells are reloaded each time a recalc is done in case one of the find/replace references was changed. Because this could be time consuming, the Show Text option will turn off this feature and only display the formula in the cell in order to speed up recalcs. In order to save time, users should turn off text while laying out their page, and turn it on when they want to view it. Another option is to hide any text cells they aren't working with. Text cells will only be reloaded if the show text option is on and the cell is visible. If only one text cell is active or visible, it won't be reloaded on recalcs, only re-displayed.

Data Cells

The format for data cells is 'filename, data_command.' Data commands are handled by the "Get Arg" routine, so at least one space must separate all values and arguments. The data commands and their functions are listed below. Tab characters or commas must separate fields in data text files and carriage returns designate the end of a record.

LOCATE — format: LOCATE literal_fieldname

Reads the first line of the data file and returns a value indicating the field number of the given field name. This is used if the first line of the data file contains the field names followed by the data. The result can then be used in the other commands below.

Example: LOCATE Address --would return 2 if Address was the second field in the first line of the data file.

FIND

format: FIND FIELD value_1 FOR FIELD value_2 logicop value

or

format: FIND RECORD FOR FIELD value_2 logicop value

In the first case, returns field value_1 for the first instance where field value_2 is true. In the second case, returns the entire record where field value_2 is true.

Examples:

FIND FIELD 2 FOR FIELD 1 > cellname
--would return the value of the second field if field number one was greater than the value contained in cellname. If cellname wasn't the name of a cell it would return field 2 if field 1 was greater then the literal string 'cellname.'

FIND FIELD cell1 FOR FIELD cell2 = Jones
--would return the field number specified by cell1 if the field number specified by cell2 was equal to 'Jones.' If Jones was the name of a cell, the field specified by cell2 would have to equal the value contained in the cell named 'Jones.'

FIND RECORD FOR FIELD 1 >= Anderson
--would return the entire record where the first field was greater than or equal to Anderson.

LIST

format: LIST FIELD value FOR FIELD value logicop value

or

format: LIST RECORD FOR FIELD value logicop value

Works the same as the FIND command except it creates a list of all values that are true.

example: LIST FIELD 3 FOR FIELD 1 > 861201
--would return a list containing all the third fields where the first field was greater than the value 861201

SUM

format: SUM FIELD value FOR FIELD value logicop value

COUNT

format: COUNT RECORD FOR FIELD value logicop value

AVERAGE

format: AVERAGE FIELD value FOR FIELD value logiop value

All of the above work as expected. These are used when the user wants a sum, count, or average from a data record, but doesn't want to load a list. It would actually be better for a user to create the list, hide it, then use the same functions available in calculated cells. It is useful, however, if the list would exceed the maximum allowable length of a data cell which is 255 characters.

Solid Cells

Solid cells are used to create bar charts. The format for solid cells is: 'max_value, actual_value[, pattern_value.' Max_value represents the maximum value for the cell and actual_value represents the actual value to represent. In other words if max-value evaluated to 100 and actual_value evaluted to 50, the cell would be filled halfway with the given pattern. If no pattern were given, the cell would be filled halfway with black.

If the cell is taller than it is wide, a vertical bar is created. If it's wider than it is tall, a horizontal bar is drawn.

Graph Cells

There are no parameters for graph cells. It simply draw a backdrop to be used for solid cells.

Pie Cells

The format for pie cells is: 'value_1, value_2, value_3,...value_20.' This simply creates a pie chart from all the values. Up to 20 values can be entered.

Graphic Cells

Graphic cells are used to draw objects. The format is: 'graphic_function, penwidth_value, penheight_value', pattern_value].' The graphic function can be any of the following:

BB	Box
FB	Filled box
RR	Rounded corner rectangle
FR	Filled rounded corner rectangle
CC	Circle
FC	Filled circle
D1	Line from upper left to lower right
D2	Line from lower left to upper right

Since the pen width and height are values, interesting effects can be created with boxes, circles, and rounded corner rectangles. Also, pen values of 0 will cause it not to be displayed. This could be useful in only having boxes or lines appear if a cell contains a value.

Pict Cells

Pict cells have no arguments. The simply hold pasted pictures from the scrapbook or clipboard. They do create files on the application's default volume (see above).

Options

Name Cell:

Used to change the name of a cell. Can be a maximum of 12 characters and can contain no spaces. Although it won't stop you from doing it, cell names should never start with numbers nor should they contain commas. If a cell name starts with a number, "Get Option" will return the value of the name rather than the value contained in the cell. If it contains commas, "Get Option" and "Get Arg" will presume that to be the end of the cell name and will find no value.

Hide Cell:

Makes the selected cell invisible. This is useful in several ways. When a cell is hidden, a user can still reference it's value but it won't be printed or displayed. Also, hidden cells are not saved as part of the Save Data option under the file menu. Therefore, if a user would like to save some, but not all, the information in a data file, they simply hide the cells they don't want included in the data file.

Under normal circumstances, users can't store anything into hidden cells. These cells are inaccesable until they're made visible again. However, script files can store into hidden cells. This makes them useful as variables for script files. For example, in creating a script that creates a merged letter, you'd create the Power Page document with the text cell that has the text, as well as calculated cells for each of the prompted entries. Then you'd hide the calculated cells. From the script file, you'd prompt the user for entries which would be stored into these hidden cells and used to merge into the letter.

Hidden cells are also useful for creating several views or input screens for a data base. Let's say you wanted to create a relational data base. One file contains a customer's locator information such as address, city, state, etc. Another file is used to track payments by each customer. You'd create the power page document with calculated cells for address, city, state, etc., as well as fields for payments. If you were going to add a new customer, you'd hide the payment fields and unhide all the locator fields, enter the information, and store it to the locator data base file. If that customer made a payment, you'd hide all the locator cells and unhide all the payment cells and store the information to the payment data base file. Although it sounds complicated, creating a script file to do this is very simple and the user simply follows the prompts.

Show Cells:

Simply unhides all cells

WYSIWYG:

Toggles on and off the what-you-see-is-what-you-get option.

Show Text:

Toggles on and off the full display of text cells.

Dollar Format:

Toggles on and off the dollar format for all calculated cells that begin with '=' in the formula.

Shift All:

Shifts all cells up, down, right, or left a given number of pixels.

Execute Script:

Executes a selected script. The script file can be on any volume or folder.

Save Date and Create Date File

These two options under the file menu create and save information to text files in the data format used by most data base applications. Only the results of calculated cells that are not hidden are saved to data files.

The Create Data File option creates a file and stores the names of all unhidden calculated cells in the file, creating a header. The names are separated by commas and end with a carriage return.

The Save Data option appends the results of all unhidden calculated cells to the selected data file. Although this is designed primarily for manipulating and storing data base information, it can also be used as a link from one power page document to another. Let's say you have a power page document that contains certain values you want included in another power page document. Simply save the data from the first power page document, then use the FIND command in a data cell in the second document to retrieve the value.

Special Tricks and Shortcuts

Cells are never recalculated until something changes. If, however, the user wants to force a recalculation, they simply select any cell, select the formula editing window and press return.

Selected cells are highlighted, even if WYSIWYG is on. In order to return to a true WYSIWYG state, the user could toggle WYSIWYG off, then back on. A faster way, however, is to click outside of any of the cells ensures no cells are selected, then click once on one of the arrows in the scroll boxes. This redraws the screen and returns it to a true WYSIWYG state.

Cells are drawn in the sequence they were created, so you should try to place down boxes, circles, etc. before you place the cells they will surround. If WYSIWYG is off, the box will cover the text and you won't see it until you turn WYSIWYG on.

You can have a maximum of 255 cells in a Power Page document. When a cell is deleted using the Clear option from the Edit menu. You don't regain that cell for use. This is necessary to ensure references don't change. Therefore, remember, if you delete a cell, it still counts agains your maximum of 255 cells.

You can create a Macintosh-like check box by creating a box, then placing a D1 graphic cell and a D2 graphic cell over the top of it. If the D1 and D2 graphics cells reference another cell for their pen width and height, the 'x' won't appear until the referenced cell contains at least a 1.