

# Contents

<b>Chapter 1: Getting started with Border</b>	<b>I</b>
<b>Installation</b>	<b>I</b>
<b>Creating a simple border</b>	<b>I</b>
<b>Saving the script</b>	<b>4</b>
<b>Adding comments to the script</b>	<b>5</b>
<b>Adding tokens and graphics</b>	<b>6</b>
<b>Using the border in PowerPoint</b>	<b>8</b>
<b>Chapter 2: Using Border</b>	<b>II</b>
<b>When to use Border</b>	<b>II</b>
<b>Starting Border</b>	<b>II</b>
<b>Border window</b>	<b>12</b>
<b>Window arrangement</b>	<b>12</b>
<b>A typical border</b>	<b>13</b>
<b>How to use Border</b>	<b>14</b>
<b>Making changes to a border</b>	<b>14</b>
<b>Dealing with error messages</b>	<b>14</b>
<b>Including PowerPoint details in a border</b>	<b>15</b>
<b>Editing in the script window</b>	<b>16</b>
<b>Creating a new border</b>	<b>16</b>
<b>Writing your own script</b>	<b>17</b>
<b>Using a drawing package</b>	<b>18</b>
<b>Other uses for a border</b>	<b>18</b>

<b>Chapter 3: Border script commands</b>	<b>19</b>
<b>Compulsory commands</b>	<b>19</b>
<b>Border script syntax</b>	<b>19</b>
<b>Command names</b>	<b>19</b>
<b>Arguments</b>	<b>20</b>
<b>List of commands</b>	<b>22</b>
<b>Summary of commands</b>	<b>38</b>
<b>Text commands</b>	<b>38</b>
<b>Margin commands</b>	<b>38</b>
<b>Box and line commands</b>	<b>39</b>
<b>Other commands</b>	<b>39</b>
<b>Compulsory commands</b>	<b>39</b>
<b>Editing keys in script window</b>	<b>40</b>
<b>PowerProject border tokens</b>	<b>41</b>

# Getting started with Border

Border is a separate program supplied with PowerProject®. You use it to design borders to frame PowerProject graphics for output. Several sample borders are supplied with PowerProject. You can use these borders as they are, or tailor them to your specific requirements.

Each border file is made up of Border Script commands, which describe the text, boxes, lines and pictures in the border.

This chapter teaches you the basics of using Border. It assumes that you have worked through the lessons in the PowerProject Concepts & Learning manual and are familiar with opening projects, starting new projects and printing from PowerProject.

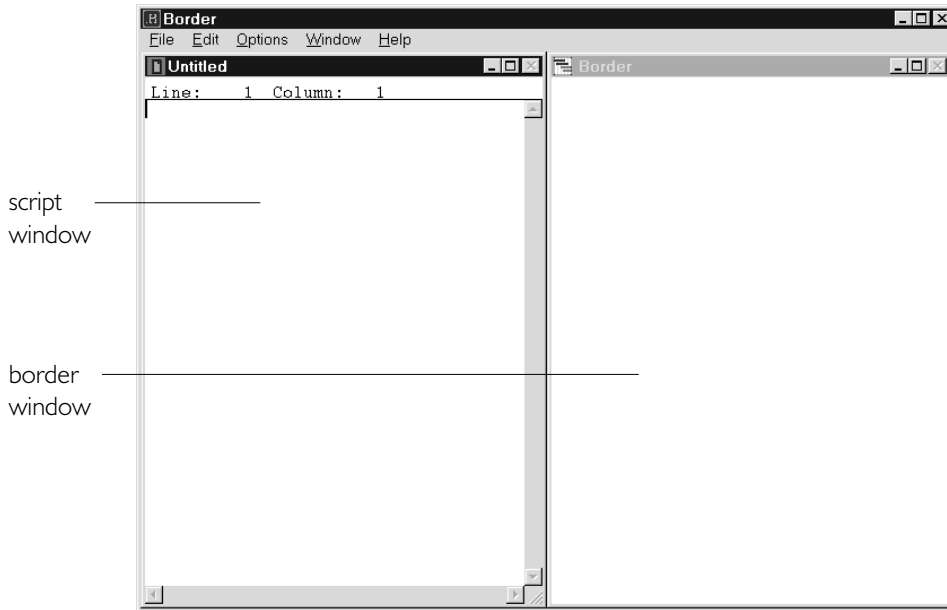
## Installation

Border is installed as part of the PowerProject installation procedure. Refer to your PowerProject Getting Started manual for details of how to install PowerProject and Border on your computer. You will need to install PowerProject and Border before you can work through the steps in this chapter.

## Creating a simple border

Start Border by selecting it from the Windows™ 95 Start menu or by double-clicking on its program icon in the Windows 3.x Program Manager. After a few moments the two main Border windows will be displayed.

A border is created using script commands which you type into the script window on the left of the screen. The border resulting from the commands is displayed in the border window on the right of the screen.



Each border must include two compulsory commands. You will create a simple border including these commands.

The cursor should be at the top left of the window, and the line at the top of the window should read `Line 1 Column 1`.

- 1 Type the following line:

```
unitsize 254
```

The first word on each line is the 'command', and the information which follows it is called the 'argument'. Press **TAB** to separate the command from the argument, so your script is easier to read. You can type in CAPITALS or lower case. This command has set the current unit of measurement to inches. If you do not use this command, the current unit of measurement will be millimeters.

- 2 Press **ENTER** at the end of the line to start a new line. On the new line, type:

```
style "A 8.5 x 11"
```

The style command is a compulsory command. It sets the name that is displayed in PowerPoint when you load the border file.

- 3 Press **ENTER** at the end of the line to start a new line. On the new line, type:

```
pagesize Letter, portrait
```

This is the second compulsory command. A standard page size and orientation follows the `pagesize` command.

Using a standard page size, Border always gives a 10 mm (0.4 inch) margin all around the paper. If you want to use a page size with no margins, you must create a custom page size and specify the width and height dimensions. For example, for Letter, portrait orientation with no margin, you would use:

```
pagesize    custom,15/2, 11
```

Now you will set the area of the page in which the chart will be printed.

- 4 Press **ENTER** at the end of the line, then type the following two lines:

```
moveto      0,1
chartarea   width-0, height-0
```

The `moveto` command sets the coordinates of the start of the following item which is the chart area. The first value is the x position and the second value is the y position. In this example, the chart area starts from the left-hand margin and 1 inch down from the top margin.

The `chartarea` command sets the extents of the chart area. Since both the width and height have been set to 0, the chart area will extend to the bottom right-hand corner of the page. The `chartarea` command is not compulsory, but you must include it to use the border with a PowerPoint chart.

Note that:

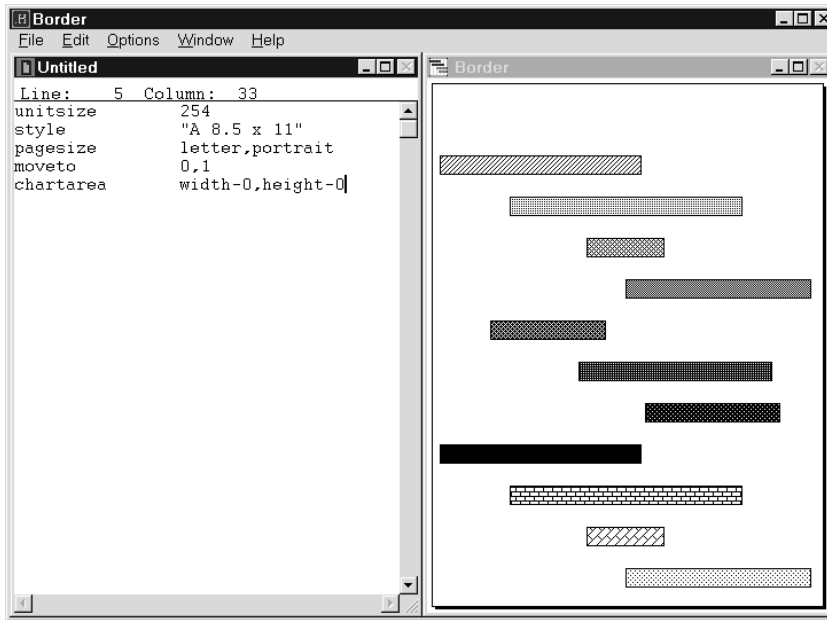
- Width is the distance between left and right page margins
- Height is the distance between the top and bottom page margins.

- 5 Press **ENTER** at the end of the `chartarea` command line. Your script should now contain the following lines:

```
unitsize    254
style        "A 8.5 x 11"
pagesize     Letter, portrait
moveto       0,1
chartarea    width-0, height-0
```

- 6 Choose **Execute** from the **Options** menu. Border now checks that you have included the compulsory commands and that the lines you have typed are correct.

Provided the script is correct, the border window should display a rectangle containing a representation of the chart. There should be a blank area across the top of the chart. Later you will edit the border so that the project title and a graphic are displayed in this area.



If any errors were found in the script, an error window will be displayed at the bottom of the screen. A message in the error window tells you which line has an error. You can **ALT** click on the error message line to go straight to the script window at the beginning of the line containing the error. Correct the line then execute the script again.

## Saving the script

At the moment, your script is called UNTITLED. You must save it as a border file before it can be used with PowerProject. Save your script now:

- 1 Select **Save As** from the **File** menu. The file selector appears.
- 2 Open your PowerProject directory (C:\WINDOWS\POWPROJ by default).
- 3 Enter the filename LESSON. All borders are given the \*.B file extension by default.
- 4 Click on **OK** to close the file selector, saving the border.

The name you have entered is displayed at the top of the script window.

Before using this border with a chart in PowerProject, we will add some comments to the script.

## Adding comments to the script

Comments in a script are useful to remind you what each section or particular line of the script is doing. In a short script you can probably tell what each line does but as your script grows, comments make it much easier to understand and amend.

To add comments to your border script:

- 1 Move the cursor to line 1, column 1 and press **ENTER** to insert a blank line.
- 2 Type:  

```
//unitsize 254 sets units to inches
```

Anything after the // will be ignored by Border, so this whole line is a comment.
- 3 Move the cursor to the end of the `chartarea` command and type:  

```
//extends to bottom right-hand corner
```

Here you have added a comment to the end of a command line. As you type, the script window scrolls horizontally to keep the cursor in view.
- 4 When you have finished typing the comment, press **HOME** to move the cursor back to the start of that line.
- 5 Select **Execute** from the **Options** menu to execute the script again with your comments. Notice that the chart representation does not change because comments do not affect the appearance of a border.
- 6 When the script has been executed, save your changes to it by selecting **Save** from the **File** menu.

## Adding tokens and graphics

Now you will learn how to include project information and graphics in a border:

- 1 Click on the line below the `chartarea` command, then type

```
moveto 1/4,1/2
```

This sets the coordinates for the next item as being one quarter of an inch from the left margin, and half an inch from the top margin page.

- 2 Add the following line below the `moveto` command:

```
textsize 1/2
```

This sets the size of the text that follows to be half an inch, or 36 points.

- 3 Below the `textsize` command, add the following line:

```
write "<proj-title>"
```

This inserts the token representing the project title at the coordinates given in the `moveto` command and at the size given in the `textsize` command. You can insert many pieces of project information into border files. A full list of the tokens available is given in the Border Help.

- 4 Press **ENTER** at the end of the `write` command line. (If you do not insert a blank line at the end of a script, an error will be reported when you execute the script.)

- 5 Select **Execute** from the **Options** menu to update the border. Notice that `<proj-title>` now appears in the top left-hand corner of the border. When you load this border into PowerProject, `<proj-title>` will be replaced by the title of the active project. You will see how to load borders into PowerProject later.

- 6 Now add the following `moveto` command to set the start coordinates of the graphic that you will include in the border:

```
moveto 5,1/4
```

This specifies that the graphic starts five inches from the left-hand margin and one quarter of an inch from the top margin.

- 7 Add this final line below the `moveto` command:

```
picture "acme.wmf",7,1
```

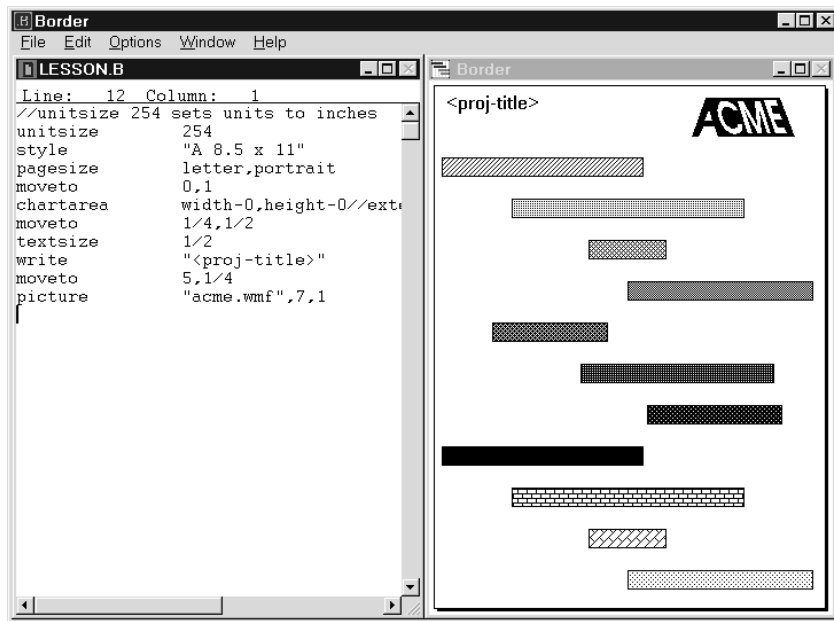


This command specifies the graphic that you want to load in the border and the coordinates at which it ends. You can see that the graphic will be 2 inches wide and three quarters of an inch high.

ACME.WMF should be in your PowerProject directory which is also where the Border program should be. If ACME.WMF and the Border program are not in the same directory, you should enter the full pathname of the graphic in the command above, e.g.:

```
picture      "c:\windows\powproj\acme.wmf "
```

- 8 Press **ENTER** at the end of the picture command line to insert an empty line at the end of the script.
- 9 Select **Execute** from the **Options** menu and notice that a graphic is now included in the top right-hand corner of the border.
- 10 Save the changes to your border by selecting **Save** from the **File** menu.

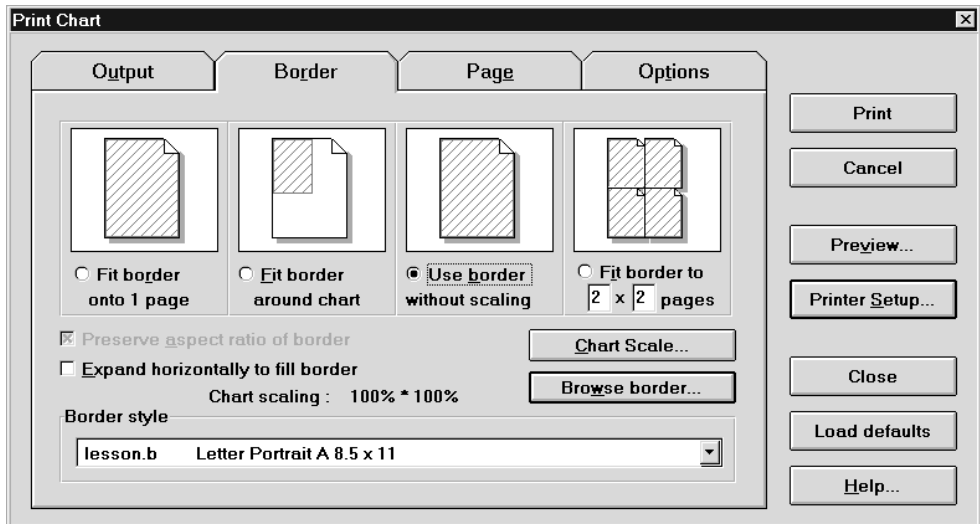


If the graphic does not appear, locate ACME.WMF on your hard disk (it should be in your PowerProject directory) and enter its full pathname in the `picture` command, then execute the border again.

## Using the border in PowerProject

Once you have saved your border, you can use it in PowerProject. This section outlines the steps involved in loading the border into PowerProject:

- 1 Run PowerProject.
- 2 Open a project if you have one available, or start a new project and draw a few bars on the chart.
- 3 Select **Print Graphics** from the **File** menu to display the tabbed **Print** dialog.
- 4 On the **Output** tab, choose a printer from the drop-down list box in the **Output to** group box and ensure that the box alongside the list box is checked.
- 5 Click on the **Printer Setup** button and ensure that Letter paper size is selected, then return to the **Print** dialog.
- 6 On the **Border** tab, click on the **Browse Border** button to locate the border you have saved.
- 7 Using the file selector that appears, select the border that you have saved then **OK** the file selector to return to the **Print** dialog.
- 8 Choose the scaling option you want. In this example, we chose **Use Border Without Scaling** so that the border fills a sheet of Letter.



- 9 Click on the **Print** button to send the chart to the current printer.

On the printed chart, you'll see that the chart occupies most of the page, with the project title in the left-hand corner and the graphic in the right-hand corner. If your project does not have a title, the word UNTITLED will appear in the position of the project title.

The following chapter provides more information about using Border. For details of a specific border script command, refer to Chapter 3 or Border's Help.



# Using Border

This chapter explains how to use the Border program. For information about a specific border script command, refer to Chapter 3: Border script commands.

## When to use Border

If you have a suitable border (saved as a .B file) you can select it from PowerProject's Print Graphics dialog as the frame for your graphic. The graphic and its border can then be printed.

- If you do not want to make any changes to a border, you do not need to use the Border program. To use an existing border to frame a PowerProject graphic, see *File – Print Graphics* in the PowerProject Reference manual.
- If you want to make changes, or create a new border, you must use Border.

## Starting Border

Start Border in the same way as any other Windows program: by selecting it from the Start menu of Windows 95, or by double-clicking on the Border icon in the PowerProject program group on your Windows 3.x desktop.

Border uses three windows. When you start Border all windows are empty. When you open a border file:

- the appearance of the border is shown in the border window
- the Border Script which creates the border is displayed in the script window
- any errors are shown in the error window.

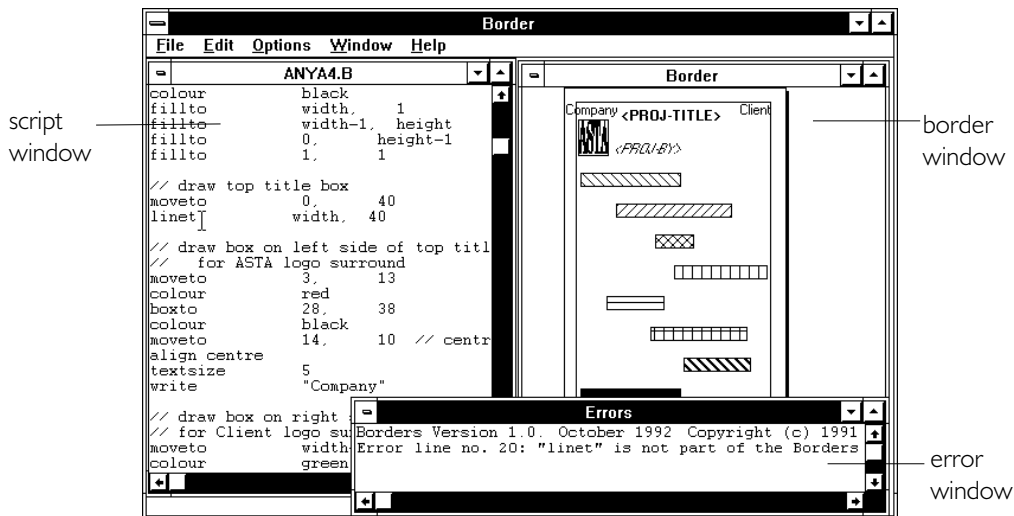
## Border window

The border window is a graphical window. The appearance of the border in this window is not quite the same as it is when you print the border chart from PowerProject:

- the chartarea (the area which is reserved for your chart) is filled with tasks representing your chart, if there are no errors
- the tokens you have included from PowerProject are displayed, not substituted by the data they represent
- the picture contained in the WMF file specified in any Picture command is not shown. Instead, the box displays the name of the WMF file.

## Window arrangement

The picture below shows a typical arrangement of windows in Border. You can change the window positions and save the window arrangement—see *Options – Save Desktop*.



## A typical border

This picture shows a typical border. Various elements of this border are explained in the examples in the following chapter.

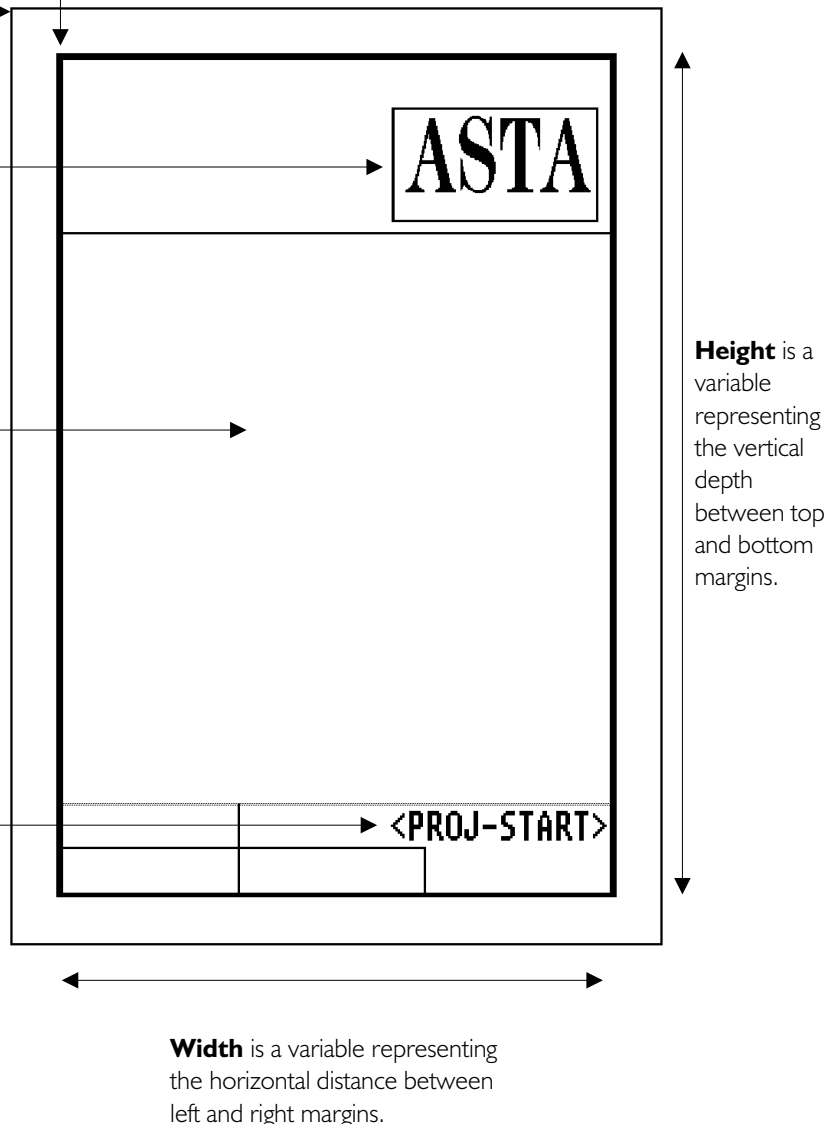
**Origin** is the name given to the top left corner of the paper. If you use a custom page size and do not set any margins, it is the same as home.

**Home** - the coordinate system starts at the top left intersection of the margins, position 0,0. This is known as Home. For any of the standard page sizes, position 0,0 is 10mm (0.4 inch) from the top and left edge of the paper.

**Picture** is a graphic image, for example a logo, which can be included in your

**Chartarea** is an area of paper within the margins. It defines the area to contain your chosen bar or filter chart.

A **token** is a piece of information from PowerProject which you want to include in the border, e.g. `<PROJ-START>` is the token for the project start date.



## How to use Border

This section provides instructions on how to make changes to a border, include PowerProject details in a border, and how to create a new border. For step by step instructions on how to use Border, work through the lesson in Chapter 1: Getting started with Border.

### Making changes to a border

There are several sample borders supplied with PowerProject. You can modify these borders to tailor them more precisely to your own needs, for example to include particular project information or to print to a particular output device (i.e. printer or plotter). By modifying these sample borders you can quickly build up a library of borders for your charts.

If you intend to modify a border, always make a copy of the border and edit the copy, leaving the original border unchanged for future use. You can copy a file in Windows or simply by opening a border file in Border and saving it under a new name using the **File – Save As** command.

Rather than remove a command line from a border script, you can comment it out by preceding it with `//` to make the command line inactive. When you have edited a border script, use the **Execute** command in the **Options** menu to check the script and display an image of the updated border in the border window. Remember to save the changes you make to a border by using the **Save** command in the **File** menu.

See *Editing in the script window* and *Dealing with error messages* for hints which are useful when you change an existing script.

### Dealing with error messages

If a border script contains any errors, they will be displayed whenever you open or execute the border. Errors are listed in the error window which is displayed along the bottom of the Border program window. If you close the error window, you can re-open it using the **View Errors** command in the **Window** menu.

If a border script does not contain any errors, then the error window cannot be displayed.



## Moving to the error line

You can move from the error window directly to an error line in the script window, using any of the following techniques:

- select the **Goto Line** command from the **Options** menu and enter the number of the line containing the error on the **Goto Line** dialog
- press **CTRL G** to display the **Goto Line** dialog, then enter the number of the line containing the error (as above)
- with the error window active, position the pointer anywhere over an error message you want to deal with, then **ALT** click
- while in the script window you can use the **CTRL N** shortcut to move to the next error line, and the **CTRL P** shortcut to move to the previous error line.

When you use any of these techniques, the script window becomes active, and the cursor is positioned at the start of the line containing the error.

The top line of the script window shows the error message from the error window instead of the normal cursor location display.

## Including PowerProject details in a border

You can include many PowerProject chart and project details in a border. You do this by choosing the token which represents the information you want, and including it your script using a `write` command. For example to print the Project title in a border, you would use the command with the token as follows:

```
write "<PROJ-TITLE>"
```

You must enclose the token in both double quotes and angle brackets, as shown.

The tokens available are listed in Chapter 3: Border script commands, and in the Help. The tokens are listed with the maximum number of characters for each token and the information they represent.

## Editing in the script window

You can edit lines in the script window in the same way that you would edit a text file in the Windows Notepad accessory. For example:

- use the arrow keys to move up, down and across lines
- press **ENTER** or **RETURN** to create a new line
- use **BACKSPACE** to delete characters to the left of the cursor
- use **DELETE** to delete characters to the right of the cursor
- use **TAB** or spaces to separate the argument from the command section of each line.

A list of the keys you can use for editing is available in the Help by selecting Keyboard shortcuts from the Help menu. A summary of the Border script syntax is also available from the Help by selecting Border Script Syntax from the Help menu.

## Creating a new border

You can create a new border by modifying an existing border, or by starting from scratch with an empty border script. If you want to start a completely new border, you can do so by:

- writing a script directly in the script window
- writing a text file using a word processing package (save the file with a .B extension)
- creating a .WMF format file with a vector drawing package, for example Corel Draw, or MS Draw, then using the **Open WMF File** command in the **File** menu
- creating a .GEM format file with a vector drawing package, for example Artline, or GEM Draw, then using the **Open GEM File** command in the **File** menu.

Whichever way you create the Border Script, you must use Border to execute the .B file containing the script, for use with PowerProject.

## Writing your own script

If you decide to write your own border script, either in the script window or using a word processor, you should draw the border on paper first. It is much easier to do this if you use graph paper. If you plan to use a standard page size, remember that Border automatically sets all the margins to 10 mm (0.4 inch).

Once you know how you want your border to look, you should consider the following:

- for accuracy and ease of reading, consider the unit size you need to use. (The default unit size is millimeters.)
- from the position of the margins and the unit size, work out a coordinate system. Remember that 0,0 is the intersection of the left and top margins. Consider using the width and height method of defining coordinates. (Coordinates are explained in Chapter 3: Border script commands.)
- you must include one `pagesize` and one `style` command in a script
- for best performance, `pagesize` should be the first command, and `style` and `chartarea` should be near the top of the script
- when you save the new border file, try to include the size and orientation in the filename, for example, USA\_A\_L.B for an A landscape border. The filename is shown on the PowerProject **Print** dialog, so a descriptive filename helps you to locate the correct border.

---

**NOTE:** If you omit the `chartarea` command, you cannot use the border to frame a PowerProject graphic, but you can use it for other purposes. See *Other uses for a border*.

---

## Script writing process

There are a number of basic steps involved in writing a border script. These steps are:

- 1 Start Border by double-clicking on the Border program icon.
- 2 To start a new script, select **New** from the **File** menu if the script window is not already empty. An untitled script window is displayed.

To modify an existing script, select **Open** from the **File** menu and choose the border that you want to edit. When you **OK** the file selector, the script will be displayed in the script window.

- 3 Type in the script, or make the changes you want to the existing script.
- 4 Check the border for errors by selecting **Execute** from the **Options** menu.
- 5 Eliminate any errors and execute the script again until it does not generate any errors.
- 6 Save the border by selecting **Save** from the **File** menu.

---

**NOTE:** You cannot save the border if it contains an error in the `style` or `pagesize` commands.

---

### Using a drawing package

You can design your borders using a drawing program. If you choose to create borders in this way, you must save the designs in your drawing package as a WMF or GEM file. To open the design in Border, use the **Open WMF File** or **Open GEM File** commands in the **File** menu. The border script will be generated automatically.

A 'rounded rectangle' graphic will be interpreted automatically as the `chartarea`. If you wish to include a `picture` command, draw a second 'rounded rectangle' and later edit this second occurrence of the `chartarea` command to the `picture` command.

The way lines and boxes are converted into Border may result in a number of redundant lines. When you become familiar with the Border commands, you will be able to edit unnecessary lines out of the script and add comment lines to make the border script more concise and easier to read.

### Other uses for a border

You can create borders without a `chartarea` command. This can be useful, for example, if you need to print several charts or histograms on a single sheet of paper for a presentation. Output the charts and histograms from PowerProject as WMF files, then create a border using the compulsory commands and several picture commands to load the WMF files. Open the border on the **Print Graphics** dialog in PowerProject and output to your printer.

By using a variety of tokens, and picture commands, you can make impressive presentation sheets and cover sheets for documents.

# Border script commands

If you decide to create a new border, or want to make changes to an existing border, you need to understand the commands which make up a border script. You can use border script commands to:

- describe the size, shape and contents of the lines, graphics, text and boxes making up a border
- specify border tokens, which represent information about PowerPoint chart and project details on the current project or a baseline of the project.

This chapter explains the border script commands and their syntax. If you use Border to convert a file created using a drawing program, all border script commands are generated automatically so you need not be aware of them.

## Compulsory commands

You must include one `pagesize` and one `style` command in a script. Although `chartarea` is not compulsory, you are unlikely to create a border for a graphic without one.

For best performance, make `pagesize` the first command and put `style` and `chartarea` near the top of the script.

You can only have one `pagesize`, `style` and `chartarea` command in a script.

## Border script syntax

The syntax of commands is simple: a command name followed by a number of arguments which are separated by commas.

```
<command name> <argument1>,<argument2>,...
```

## Command names

Case is not relevant when typing Border command names. For example, you could type `topmargin`, `TopMargin` or `TOPMARGIN`. The format Border uses when it generates commands are shown in this section.

## Arguments

The arguments take different forms. Depending on the command, they can be text, numbers, coordinates, variables or expressions.

### Text

Text is enclosed by double quotes, for example:

```
Write "Produced by MIS"
```

### Numbers

Numbers are entered in the current unit size, which is millimeters unless you have used the `unitsize` command.

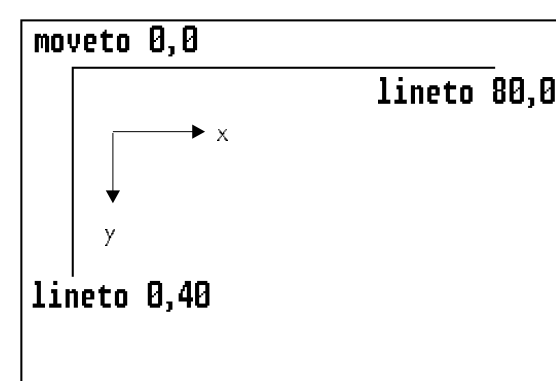
---

**NOTE:** The examples in this chapter shown the current unit size as millimeters.

---

### Coordinates

Coordinates must be entered in the current unit size. The coordinate system starts at the intersection of the top and left margins, known as home (0,0). You should set the margins before using any commands with coordinates. The diagram below shows an example of how to enter coordinates, using the `lineto x, y` command.



## Variables

Border provides you with two variables which enable you to make borders suitable for use on different paper sizes, or for different orientations.

They are 'height'—the vertical depth between top and bottom margins, and 'width'—the horizontal distance between left and right margins. See *A typical border*, in Chapter 2: Using Border.

## Expressions

You can include mathematical operators with a variable, to give an expression. The operators you can use are:

- Minus           —
- Plus            +
- Divide         /
- Multiply        \*

For example, you could center 'Management Summary' using the commands

```
align          centre
//center point between margins, 2 cm from bottom
moveto        width/2, height-20
Write         "Management Summary"
```

## Comments

You can add comments to make a border script easier to understand. Comments can be on a line of their own, or at the end of a line containing a command. Wherever you add a comment, you must start with //, for example:

```
//Draw a box on the paper
```

or

```
Boxto ... // Draw a box on the paper
```

## List of commands

Commands are listed in alphabetical order in this section. All commands are described as though you were controlling the movement of a pen across a sheet of paper, although the output may be on printer or plotter.

At the end of the section are tables which group the commands according to their function.

### align

This command controls the alignment of text you supply with the `write` command. If you use `align left`, text is left justified—it is written across the page from left to right from the current pen position. You can change the setting to `right` if you want the text to be right flush, or `centre` for text to be centered on the current pen position. You can use the `align` command anywhere in the script before the `write` command.

#### Format

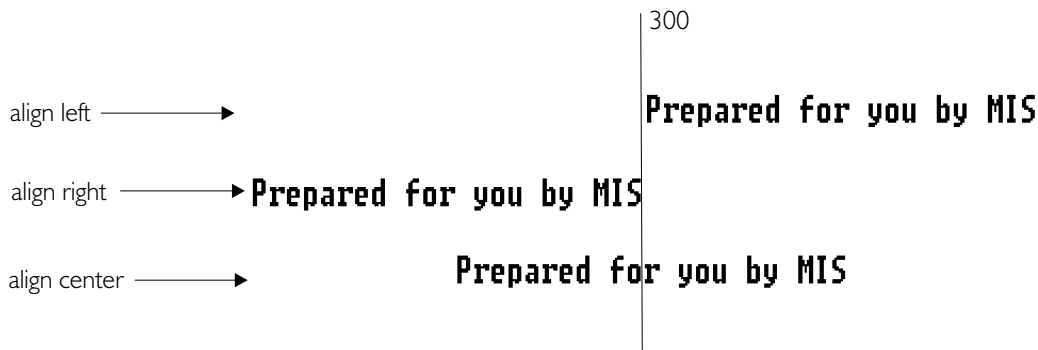
```
align left | right | centre
```

#### Default

The default is align left.

#### Example

```
moveto 300,300
align left
write "Prepared for you by MIS"
```





## baseline

Makes border tokens refer to a baseline. The default is that all border tokens refer to the current project. Specify the baseline you want the tokens to refer to, according to the order in which the baselines were created.

The earliest baseline is the first baseline that was created from the project and the latest is the most recent baseline. Once you have used this command, all border tokens refer to the specified baseline until you specify that they should refer to the project.

### Format

```
baseline earliest | latest <baseline number>
```

### Default

If you do not use this command, all tokens refer to the current project. If you do not specify a baseline number, PowerProject uses 1 to refer to the earliest or latest baseline.

### Example

To make the token "<pdb-filename>" refer to the 3rd baseline created for a project

```
baseline    earliest 3
write      "<pdb-filename>"
```

To make tokens refer to the most recent baseline

```
baseline    latest
```

To specify that tokens should no longer refer to a baseline, but to the current project, use the `project` command or use the baseline number 0.

```
baseline earliest 0 or baseline latest 0
```

### Associated commands

```
project
```

## **bold**

Turn bold text on or off. The default is off, so if you want text in bold, you can turn it on. At the end of bold text, turn it off.

### **Format**

`bold     on | off`

### **Default**

Bold is off until you turn it on.

### **Example**

To produce **Prepared for you by MIS**

```
bold     on
write    "Prepared for you by MIS"
bold     off
```

## **bottommargin**

Set the height of the bottom margin. If you are printing on a printer or plotter which has mandatory margins, you must use this command to make a bottom margin to ensure your border is on a printable part of the page. Check your printer manual to find out if your printer expects a minimum margin.

### **Format**

`bottommargin <height>`

### **Default**

If you do not use this command, for custom page size the bottom margin is set to zero. For the standard page sizes, the bottom margin is set to 10 mm (0.4 inch).

### **Example**

To set a bottom margin of 10 mm (0.4 inch):

```
bottommargin        10
```

## boxto

Draw a hollow box from the current pen position to the coordinates you supply. Before using **boxto** you should move the current pen position to the start of the box, for example the top left corner, using **moveto**.

You can also use the variables 'height' and 'width' to draw a box relative to the paper width.

After drawing the box, the pen position is set to the coordinates you supplied. For example, if you start the box at the top left hand corner, the pen finishes at the bottom right corner of the box.

### Format

```
boxto <rightedge(x)>,<bottomedge(y)>
```

### Default

No default: you must supply the coordinates.

### Example

To draw a box 10mm from the top and left margins which is 20 mm square

```
moveto 10,10
boxto 30,30
```

To draw a box on the margins on any paper size

```
moveto 0, 0 //Home
boxto Width, Height
```

### Associated commands

```
moveto, color, unitsize
```

## chartarea

This is a key command in a Border Script. Use it to define the part of the page onto which PowerProject should place your graphic (chart, histogram or calendar). If the **chartarea** command is not included in your script, PowerProject has nowhere to place a chart in the final output.

The `chartarea` is a rectangle starting from the current pen position, ending according to the coordinates you supply. It must be within the margins. You can use the variables 'height' and 'width' to set the `chartarea`, but if you do, the `pagesize` command must precede `chartarea`.

After setting the `chartarea`, the pen position is set to the coordinates you supplied.

### Format

```
chartarea <rightedge(x)>,<bottomedge(y)>
```

### Default

None. Although this command is not compulsory, if you save a border without a `chartarea`, you should have a specific reason for doing so because PowerProject cannot use it as a border for a chart. You might omit the `chartarea` command for example, if you want to produce a cover page for your charts containing project or chart details using tokens.

### Example

The examples show `chartarea` set using the 'height' and 'width' variables, since the border is then more flexible. However, you can measure the exact size of the `chartarea`, and supply these figures as coordinates.

To set a `chartarea` which is 1 mm from the left and right margins, 4 cm below the top margin and 2 cm above the bottom margin,

```
moveto 1, 40 //1mm from left, 4cm from top
chartarea width-1, height-20
//1mm from right, 2 cm from bottom
```

To set a chart area which is the full width of the page, leaving an area of 10 cm at the bottom

```
moveto 0, 0,
chartarea width, height-100
```

## color

Select the pen color to be used for all subsequent drawing commands. You can change color as often as you like. The sixteen colors you can choose from are:

black	blue	darkgrey	darkblue
cyan	green	darkcyan	darkgreen
magenta	red	darkmagenta	darkred
white	grey	yellow	darkyellow

### Format

`color <color> or colour <colour>`

### Default

If you do not specify a color, the default is black

### Example

To change the color to darkred

```
color    darkred
write    "Prepared for you by MIS"
```

## fillto

`Fillto` is similar to the `boxto` command, but it produces a solid box, coloured in the current color setting—default black.

You can use this command to produce a thick line by making a long thin box. You can also use it to fill any part of the border or chart area to enhance the presentation of a chart.

After drawing and filling the box, the pen position is set to the coordinates you supplied.

### Format

`fillto <rightedge(x)>,<bottomedge(y)>`

### Default

No default: you must supply the coordinates.

**Example**

To draw a red line, 1 mm thick, just inside the margins

```
moveto 0,          0          //Home
colour red
fillto width,      1
fillto width-1,    height
fillto 0,          height-1
fillto 1,          1
```

**Associated commands**

moveto, color, unitsize

**italic**

Turn italic on or off. The default is off, so if you want text printed in italics, you must turn it on. At the end of the italic text, you must turn it off.

**Format**

Italic on | off

**Default**

Italic is off until you turn it on.

**Example**

To produce *Prepared for you by MIS* :

```
italic on
write "Prepared for you by MIS"
italic off
```

**leftmargin**

Set the width of the left margin. If you are printing on a printer or plotter which has mandatory margins, you must use this command to make a left margin to ensure your borders are on a printable part of the page. Check your printer manual to find out if your printer expects a minimum margin.

Home—coordinate 0,0—is the intersection of the top and left margins. You should set the margins at the beginning of a border script. Once you have set them, and established your coordinate system, you should not change their positions.

### Format

```
leftmargin <width>
```

### Default

If you do not use this command, for custom page size the left margin is set to zero. For the standard page sizes, the left margin is set to 10 mm.

### Example

To set a left margin of 10mm (0.4 inch):

```
leftmargin 10
```

## lineto

Draw a line in the selected color from the current pen position. You supply the length and direction of the line by supplying the coordinates for the end pen position. After drawing the line, the pen position is set to the coordinates you supplied.

### Format

```
lineto <right(x)>,<bottom(y)>
```

### Example

To draw a line along the top margin

```
moveto 0, 0
lineto width, 0
```

### Associated commands

```
moveto, color, unitsize
```

## moveto

Move the pen to the location supplied by the coordinates. The pen position is in *current unit size*. You need to use this command to position the pen *before* `chartarea`, `picture`, `write`, `lineto`, `boxto` or `fillto`.

**Format**

```
moveto <x-position>,<y-position>
```

**Default**

The pen position is initially set to 0,0

**Example**

To locate the pen 1 mm from the left margin and 40 mm below the top margin:

```
moveto 1, 40
```

**Associated commands**

```
boxto, fillto, lineto, picture, write
```

**pagesize**

You must include this command in a border script, to set the page size.

There are two ways you can set the page size: by supplying a page type and orientation from the list below, or by creating a custom page type by supplying the page's height and width in current unit size.

Whichever method you choose, the 'height' and 'width' variables are automatically set according to your **pagesize** command. You must set the pagesize before using variables.

Page types recognized by Border are:

A0	A1	A2	A3	A4	A5	B5
Broad	Half	Ledger	Legal	Letter	Wide	

Page orientation: landscape—short side of paper vertical,  
or: portrait—long side of paper vertical.

**Format**

```
pagesize <pagetype>, portrait | landscape
```

or

```
pagesize custom,<pagewidth(x)>,<pageheight(y)>
```



## Default

There is no default for `pagesize`—you must include this command.

## Example

To set the page size to the standard A size, portrait orientation

```
pagesize      A,          portrait
```

To set the page size to 100 mm square

```
pagesize      custom,    100,      100
```

## picture

Load a graphics image from a .WMF or .BMP file into a rectangle drawn from the current pen position and ending according to the coordinates you supply.

---

**NOTE:** Some Windows 3.0 printer and plotter drivers will not print BMP files. So ensure that you have the latest driver for your printer or plotter.

---

The image, for example a logo, is scaled to fit the given rectangle.

After the `picture` command, the pen position is set to the coordinates you supplied.

When you Execute the border, the border window shows the logo in the desired location.

```
moveto 20,50
```



```
picture "JUSTASTA.WMF",120,150
```

To ensure that your graphic is the right way up, first `moveto` the top left hand corner of the area where you want to position your graphic file. Use the `picture` command to supply the .WMF or .BMP file name and the bottom right hand coordinates.

**Format**

```
picture "<filename>" ,<rightedge(x)>,<bottomedge(y)>
```

**Default**

None

**Example**

To load a file called LOGO.BMP into a 30 mm square box

```
moveto 0,0
//go to the top left hand of the box

picture "C:\ARTWORK\LOGO.BMP" ,30,30
//draw the picture

moveto 0,0
//return to top left hand coordinate for the box

boxto 30,30
//draw a line to surround the logo
```

---

**NOTE:** It is recommended that the full path name of .WMF and .BMP files are included in a border file, to prevent problems during output.

---

**project**

Makes border tokens refer to the current project. Use this command if you have previously made all tokens refer to a baseline of the current project, using the **baseline** command.

**Format**

This command has no arguments.

**Default**

The default is that border tokens refer to the current project.

**Example**

To write out the end date of the first baseline and underneath it write out the end date of the current project

```

textsize      10
moveto        0,0
baseline      earliest
write         "<Proj-end>"
project
moveto        0,10
write         "<Proj-end>"

```

## rightmargin

Set the width of the right margin.

If you are printing on a printer or plotter which has mandatory margins, you must use this command to make a right margin to ensure your borders are on a printable part of the page. Check your printer manual to find out if your printer expects a minimum margin.

### Format

```
rightmargin <width>
```

### Default

If you do not use this command, for custom pagesize the right margin is set to zero. For the standard pagesizes, the right margin is set to 10 mm.

### Example

To set a right margin of 10 mm (0.4 inch):

```
rightmargin 10
```

## style

You must use this command to give the border a name. It is not printed on the border, but is displayed by PowerProject on the **Print Graphics** dialog, where you can select the border to be used to surround your output.

### Format

```
style "<freetext>"
```

### Default

There is no default. Every script must include the style command.

### Example

```
style "A Portrait Border for Acme Supplies"
```

## textsize

Set the height of text characters in current unit size.

Border allows you to enter any number, and text is displayed in the size you have indicated. However, you must select a size which can be produced by your output device. If you are using a plotter or Postscript printer, any size can be matched exactly. For output on all other printers, you should limit the text sizes you choose to those provided by the printer's fonts. If you use text sizes which are not available, PowerPrint chooses the closest size available.

### Format

```
textsize <Characterheight>
```

### Example

To set the height of characters to 10 mm, which would need a large printer font of 28 point.

```
textsize 10
```

The character height (in mm) to use for the commonest point sizes:

mm	Point	mm	Point
2.1	6	4.9	14
2.8	8	5.6	16
3.5	10	6.3	18
4.2	12	7.1	20

## **topmargin**

Set the height of the top margin in current unit size.

If you are printing on a printer or plotter which has mandatory margins, you must use this command to make a top margin to ensure your borders are on a printable part of the page. Check your printer manual to find out if your printer expects a minimum margin.

The intersection of the top margin and left margin is the start of the coordinate system—home—0,0.

### **Format**

```
topmargin <height>
```

### **Default**

If you do not use this command, for custom pagesize the top margin is set to zero. For the standard pagesizes, the top margin is set to 10 mm.

### **Example**

To set a top margin of 20 mm (0.8inch):

```
topmargin 20
```

## **underline**

Turn underline on or off. The default is off, so if you want text underlined, you must turn it on. At the end of the text you want to underline, you must turn it off.

### **Format**

```
underline on | off
```

**Default**

Underline is off until you turn it on.

**Example**

To produce Prepared for you by MIS:

```
underline on
```

```
write "Prepared for you by MIS"
```

```
underline off
```

You can set more than one attribute for text, for example to have the text in bold, and underlined—Prepared for you by MIS

```
underline      on
bold           on
write          "Prepared for you by MIS"
bold           off
underline      off
```

**unitsize**

Use this command to change the units you use to supply numbers and coordinates. You supply the size of the unit you want to use, in 1/10s of a millimeter.

**Format**

```
unitsize <newsize>
```

**Default**

Unless you use this command to change current units, the current unit size is millimeters.

**Examples**

To change to inches and move the pen

```
//Give measurements in inches (254*1/10mm = 1")
unitsize      254
```

```
//Move 1" across and 1" down page from origin
moveto        1, 1
```

To change to centimeters

```
unitsize          100
//Give measurements in centimeters
```

## **write**

Write the text you supply in double quotes at the current pen position. Use this command to place the text you supply anywhere within the margins. You can use insert project details into a border by supplying a token name in double quotes.

First move the pen to the required start position using `moveto`. If you want to change color or start underlining or writing in bold, do so before using the `write` command. You should also use the `align` command before `write`.

When the text has been written, the pen returns to the start position.

### **Format**

```
write "<freetext>"
```

### **Default**

none

### **Example**

To write a string of text:

```
write "Prepared for you by MIS"
```

To include the start date taken from your chart, use a token

```
write "Chart Start Date: <CHART-START>"
```

### **Associated commands**

`align`, `bold`, `color`, `italic`, `moveto`, `textsize`,  
`underline`

## Summary of commands

### Text commands

<code>moveto</code>	<code>&lt;x value&gt;,&lt;y value&gt;</code>
<code>align</code>	<code>left   right   centre</code>
<code>bold</code>	<code>on   off</code>
<code>color</code>	<code>black   blue   cyan   green</code> <code>magenta   red   white   yellow</code> <code>grey   darkgrey   darkblue</code> <code>darkcyan   darkgreen   darkmagenta</code> <code>darkred   darkyellow</code>
<code>italic</code>	<code>on   off</code>
<code>textsize</code>	<code>&lt;value&gt;</code>
<code>underline</code>	<code>on   off</code>
<code>write</code>	<code>"&lt;text&gt;"</code>

### Margin commands

<code>bottommargin</code>	<code>&lt;value&gt;</code>
<code>topmargin</code>	<code>&lt;value&gt;</code>
<code>leftmargin</code>	<code>&lt;value&gt;</code>
<code>rightmargin</code>	<code>&lt;value&gt;</code>



## Box and line commands

<code>moveto</code>	<code>&lt;x value&gt;,&lt;y value&gt;</code>
<code>boxto</code>	<code>&lt;x value&gt;,&lt;y value&gt;</code>
<code>fillto</code>	<code>&lt;x value&gt;,&lt;y value&gt;</code>
<code>lineto</code>	<code>&lt;x value&gt;,&lt;y value&gt;</code>

## Other commands

<code>unitsize</code>	<code>&lt;value&gt;</code>
<code>chartarea</code>	<code>&lt;x value&gt;,&lt;y value&gt;</code>
<code>picture</code>	<code>"&lt;filename.WMF&gt;",&lt;x value&gt;,&lt;y value&gt;</code>
<code>baseline</code>	<code>earliest   latest &lt;baseline number&gt;</code>
<code>project</code>	<code>no arguments</code>

## Compulsory commands

<code>style</code>	<code>"&lt;text&gt;"</code>
<code>pagesize</code> <code>or</code> <code>pagesize</code>	<code>custom,&lt;x value&gt;,&lt;y value&gt;</code>  <code>A0   A1   A2   A3   A4   A5   B5</code> <code>broad   half   ledger   legal</code> <code>letter   wide</code>  <code>,portrait   landscape</code>

## Editing keys in script window

<b>HOME</b>	Start of line
<b>END</b>	End of line
<b>PGUP</b>	Page up
<b>PGDN</b>	Page down
<b>CTRL →</b>	Word right
<b>CTRL ←</b>	Word left
<b>CTRL HOME</b>	Start of file
<b>CTRL END</b>	End of file
<b>+</b> (on numeric keypad)	Copy line to Clipboard
<b>–</b> (on numeric keypad)	Paste line to Clipboard
<b>INS</b>	Insert line from Clipboard
<b>CTRL G</b>	Goto line...
<b>CTRL N</b>	Next error
<b>CTRL P</b>	Previous error

### In error window

<b>ALT</b> click	Go to error line in script window
------------------	-----------------------------------

## PowerProject border tokens

All the items of information which you can extract from PowerProject and include in a border are shown in the table below.

Items which you can edit in PowerProject are marked by a \*. Most are on the **Project** or **Chart** tabs of the **Details** dialog.

ITEM	TOKEN	MAX SIZE
<b>Project items</b>		
*Project title	PROJ-TITLE	40
*For (Client)	PROJ-FOR	40
*By (Company)	PROJ-BY	40
*Project start date	PROJ-START	13
Project end date	PROJ-END	13
Project report date	PROJ-REP-DATE	13
Project % complete	PROJ%COMPLETE	3
Project elapsed duration	PROJ-DURATION	27
*Chart title if root or not	CHART-TITLE	40
*Chart title if not root	CHART-SUBTITLE	40
*Pathname—no project name	CHART-PATH	variable
<b>Chart items</b>		
*Chart comment	CHART-COMMENT	40
*Chart author	CHART-AUTHOR	32
*Chart issue date	CHART-ISSUE-DATE	13
*Chart revision comment	CHART-REV-COMMENT	40
*Chart programme number	CHART-REV-NUM	10
or	CHART-PROGRAMME	10
*Chart revision date	CHART-REV-DATE	13
*Chart manager	CHART-MANAGER	32
*Chart revision	CHART-REVISION	3
Chart start date	CHART-START	13
Chart end date	CHART-END	13

ITEM	TOKEN	MAX SIZE
Chart work content	CHART-WORK-CONTENT	27
Chart elapsed duration	CHART-DURATION	27
Chart % complete	CHART%COMPLETE	3
*Chart report date	CHART-REP-DATE	13
Number of bars	CHART-NUM-BARS	3
Number of segments	CHART-NUM-SEGS	5
Number of local links	CHART-NUM-LOCLINKS	5
Number of X-chart links in	CHART-NUM-XLINKS-IN	5
Number of X-chart links out	CHART-NUM-XLINKS OUT	5
Number of X-chart links total	CHART-NUM-XLINKS	5
<b>Other items</b>		
Project database filename	PDB-FILENAME	12
Current date (of output)	DATE	13
Current time (of output)	TIME	5
Metafile name	WMF-FILENAME	12
Border file name	BORDER-FILENAME	12
*Issue date of baseline	BSLN-ISSUE-DATE	13
*Summary of baseline	BSLN-SUMMARY	40
Comment about baseline	BSLN-COMMENT	320