



# PartitionMagic Pro Scripting

*Click a link below to view the following information:*

- [Using Script Processing](#)
- [Scripting Syntax](#)
- [Scripting Commands](#)
- [Script File Statements](#)
- [Operators](#)
- [Queries](#)
- [Script Suggestions and Notes](#)
- [Sample Scripts](#)

## Using Script Processing

You can use PowerQuest® PartitionMagic® Pro scripts to make changes to the partitions and volumes on a machine. A script is an ASCII text file with text statements that define the operations you want to perform. You can create a script file with ScriptBuilder or with any text editor.

### Creating a New Script

- 1 From PartitionMagic, click **Tools** ► **Scripting** ► **ScriptBuilder**.



- 2 From ScriptBuilder, click **File** ► **New**.
- 3 Click **Insert**, then select a command, query, or statement.

Some commands that you insert also have parameters that you can specify.

For example, when you insert a Create command (such as Create /FS=FAT) into the script edit pane, you click **Insert** ► **Command** ► **Create**, then select an option from the menu. The Create command, among many other commands in ScriptBuilder, has a variety of associated parameters that let you specify more information beyond just the inserted command. In this example, the Create command includes parameters that let you specify the partition label, size, and position on the hard disk.

Therefore, to add parameters to a command that is inserted in the script edit pane, click the line that you want to affect, click **Insert** ► **Parameter**, then select a

parameter option. If you click Insert and the Parameter option is dimmed (unavailable), the command does not have any parameters.

To replace an incorrect parameter in a command, select (highlight) the parameter syntax, click **Insert ► Parameter**, then choose a new parameter.

- 4** In the script edit pane, specify the options you want.
- 5** Repeat steps 3 and 4 until you are finished creating the script.
- 6** To save your script file, click **File {bmc 1Step.bmp} Save As**, type a path and filename for the file, then click OK.

The script text file allows comments in the C++ form (`//`). These can be either a full line or after a valid statement.

To delete syntax, select the line you want to remove in the script edit pane, then press <Delete>. You can also right-click in the script editing pane to cut, copy, paste, or delete lines from the script.

When you are finished creating and saving the script file, you are ready to use the **Play** command on the Script pull-down menu to compile the file. If the script has not yet been saved, the **Play** command changes to **Save and Play**.

### ***Checking Script Syntax***

Use Check Syntax to test the entire syntax in the script for errors.

- 1** From ScriptBuilder, click **Script ► Syntax Check**.

Any errors in the syntax are listed in the error pane directly below the script edit pane. You can increase or decrease the viewing area of either pane by dragging the horizontal separator bar up or down.

### ***Recording Scripts***

Instead of creating a script by selecting items from the **Insert** menu in ScriptBuilder, you can “record” a script while you use PartitionMagic.

- 1** From PartitionMagic, click **Tools ► Scripting ► Record**.
- 2** Select operations in PartitionMagic as you normally would.

Some operations may be dimmed (unavailable) while you are recording a script. Dimmed operations are not recordable in a script.

When you select an option, the associated script command is sent to ScriptBuilder and inserted into the script edit pane at the current cursor location.

- 3 When you are finished recording, click **Tools ► Scripting ► Record** to turn off recording.
- 4 You may be prompted to discard any virtual changes you made to your machine while recording the script. If so, click **Yes** to erase all changes, or click **No** to apply all changes.

Regardless of whether you click Yes or No, your script will remain intact.

- 5 Click **File ► Save** or **File ► Save As** to save the script file.

### ***Playing Scripts***

You can play a script from ScriptBuilder, PartitionMagic Pro, or the PartitionMagic Pro command line.

<b>To play from:</b>	<b>Do this:</b>
ScriptBuilder	Click <b>Script ► Play</b> or <b>Save and Play</b> .
PartitionMagic Pro	Click <b>Tools ► Scripting ► Play Script File</b> , specify the path and filename of the .PQS file you want to run, then click <b>Play</b> .
PartitionMagic Pro command line	Use the /CMD command line switch to run a script from the command line without opening PartitionMagic. If the script file were named SCRIPT.TXT, the syntax for running the program from the script would be:  <code>PQMAGIC /CMD=SCRIPT.TXT</code>

If the program encounters an error, the script will end immediately and an error will be displayed in the message pane. You can also determine if an error has occurred and what error it was by looking at the log file or error file. These two files are only created if you specify them on the command line.

Sometimes the script cannot complete an operation because the disk configuration is not correct for the script commands. If that is the case, it reports “0 operations pending” and “script complete.” No error is reported.

Each operation in a script is performed on the partition that was last specified. You must specify the correct partition before running a script.

## Scripting Syntax

Several special characters are used when describing the syntax of script file statements. These are described below. Script file statements are not case-sensitive. Before performing an operation, you must first select the drive and partition that you wish to act upon.

{ } - Denotes a required parameter

[ ] - Denotes an optional parameter

| - Denotes a choice among two or more options

## Scripting Commands

You can use scripting commands to perform almost any operation that is available from the **Operations** menu in PartitionMagic.

You insert a command into the script edit pane of ScriptBuilder by clicking **Insert ► Command**.

To add any optional or required parameters to a command that is inserted in the script edit pane, click the line that you want to affect, click **Insert ► Parameter**, then select a parameter. If you click **Insert** and the **Parameter** option is dimmed (unavailable), the command does not have any parameters.

### ***Allow Manual Reboot***

Allow the script to run even if it is determined that the program cannot reboot the machine remotely, after changes are made. This should be the first statement if used.

No parameters.

### ***Bad Sector Retest***

Retest the current partition for bad sectors and unmark any bad sectors that have been set incorrectly.

### ***Check***

Use to check a selected partition for errors.

No parameters.

You should use the check command in almost every script for each partition you will be affecting. If any disk error exists, the check command will detect it and exit the script before any problems occur.

## ***Cluster Analyzer***

### **Syntax**

```
Cluster Analyzer [/SetClusterSize={ Recommended|512 | 1 | 2  
| 4 | 8 | 16 | 32 | 64 }]
```

Get Cluster Analysis information about a particular partition.

If this operation is used with out any parameters it will output a Cluster Analysis screen.

<b>Parameter</b>	<b>Description</b>
/SetClusterSize	Changes the cluster size to the specified size.

### ***Convert To FAT***

Convert a FAT32 or NTFS partition to FAT.

### ***Convert To FAT32***

Convert a FAT or NTFS partition to FAT32.

### ***Convert To HPFS***

Convert a FAT partition to HPFS.

### ***Convert to NTFS***

Convert a FAT partition to NTFS. Under Windows® 2000, you can convert a FAT32 partition to a NTFS partition. If you use this command, you must be in Windows.

### ***Convert To Primary***

Convert a logical partition to a primary partition.

### ***Convert To Logical***

Convert a primary partition to a logical partition.

## Copy

### Syntax

Copy [/Position = {Beginning | End}]

Parameter	Description
/Position	<i>(Optional)</i> Must be followed by either END or BEGINNING to specify where the partition will be copied in the unallocated space. It will default to the beginning.

Copy a partition to unallocated space. For the Copy command to work correctly, a disk and partition need to be selected and a destination disk and unallocated space need to be selected.

The Copy command should be preceded by the following commands:

- Select Disk {Num}
- Select Partition {PartitionLetter | "Volume Label" | Extended | Next | Previous | Num}
- Select Destination Disk {Num}
- Select Destination Unallocated {Num | First | Last | Largest | After Partition Num | Before Partition Num | Next | Previous}

## Create

### Syntax

```
Create /FS={FAT | FAT32 | HPFS | LINUXEXT2 | LINUXSWAP | NTFS  
| EXTENDED | UNFORMATTED} [/Label="NEW LABEL"] [/Size=Value]  
[/Position={BEGINNING | END}]
```

Create a new partition, and, optionally, format it.

Parameter	Description
/FS	<i>(Required)</i> It can be any of the above specified strings. There may be cases where creating with a certain /FS would fail. For example, trying to create an Extended partition when one already existed.

<b>Parameter</b>	<b>Description</b>
<i>/Label</i>	<i>(Optional)</i> Replace "NEW LABEL" with the desired volume label. It must be 11 characters or less for FAT partitions.  Labels must be 16 characters or less for Linux Ext2 partitions and 32 characters or less for NTFS partitions. The label must be in double quotes. The script may fail if invalid characters are entered.
<i>/Size</i>	<i>(Optional)</i> Specified in megabytes and will default to the size of the unallocated space if not specified.
<i>/Position</i>	<i>(Optional)</i> Must be followed by either END or BEGINNING to specify where the partition will be created in the unallocated space. It will default to the beginning.

## **Delete**

### **Syntax**

```
Delete {Volume Name | "NO NAME" | "SWAPSPACE2" | "UNKNOWN"} [ /Shred ]
```

Use to delete a selected partition.

<b>Parameter</b>	<b>Description</b>
Volume Name	You must enter the volume name to ensure that you are destroying data in the proper partition only. The value entered must always be preceded and followed by double quotes. If the partition label is blank and the partition is FAT or HPFS, enter "NO NAME." If the partition is not FAT or HPFS, enter "UNKNOWN." To delete an unformatted partition or an extended partition, enter "NO NAME." To delete a Linux swap partition, enter "SWAPSPACE2."
<i>/Shred</i>	<i>(Optional)</i> Permanently destroys both the data and the selected partition. Undelete can not be used to reverse this command. This command takes much longer than Delete.  The Shred parameter functions the same as the "Delete and Secure Erase" operation found in PartitionMagic.

## **Delete All**

### **Syntax**

Delete All [/Shred]

Use to delete all partitions from the selected disk.

## **Format**

### **Syntax**

```
Format {Volume Name | "NO NAME" | "UNKNOWN" } /FS={ FAT |  
FAT32 | HPFS | LINUXEXT2 | LINUXSWAP | NTFS } [/Label="NEW  
LABEL"]
```

Use to format a selected partition.

<b>Parameter</b>	<b>Description</b>
Volume Name	Required to format a partition UNLESS the partition is either an extended partition or an unformatted partition. This is a check to ensure that you are destroying data in the proper partition only. The value entered must always be preceded and followed by double quotes. If the partition label is blank and the partition is FAT or HPFS, enter "NO NAME" as the label. If the partition is not FAT or HPFS, enter "UNKNOWN" as the label.
/FS	<i>(Required)</i> Can be any of the above specified strings. There may be cases where formatting with a certain /FS would fail, for example trying to format a FAT partition past 1024 cylinders.
/Label	<i>(Optional)</i> Replace "NEW LABEL" with the desired volume label. It must be 11 characters or less for FAT partitions. Labels must be 16 characters or less for Linux Ext2 partitions and 32 characters or less for NTFS partitions. The label must be in double quotes. The script may fail if invalid characters are entered.

## **Hide**

Hide the currently selected partition. See also, "Unhide."

## **Info**

### **Syntax**

```
Info [/Usage ] [/Waste ] [/Partition ] [/FS]
```

Use to get information about a selected partition. All parameters are optional. When no parameters are supplied, the default is to show information for all that apply.

<b>Parameter</b>	<b>Description</b>
<code>/Usage</code>	<p>The Disk Usage screen is available for the FAT, FAT32, NTFS, and HPFS file systems.</p> <p>This screen shows you the following information in bytes, megabytes, and as a percentage:</p> <ul style="list-style-type: none"><li>• Used space on the partition, including space wasted by clusters</li><li>• Unused space on the partition</li><li>• Bad space on the partition</li><li>• Total space on the partition (found by adding the three previous lines)</li></ul>
<code>/Waste</code>	<p>The Cluster Waste screen applies only to partitions that use either the FAT or FAT32 file system. This screen shows the following:</p> <ul style="list-style-type: none"><li>• Current cluster size in bytes or kilobytes</li><li>• Data stored on the partition in bytes and megabytes</li><li>• Wasted space on the partition in bytes and megabytes</li></ul> <p>Total Used space in bytes and megabytes (found by adding the numbers on the two previous lines)</p>

Parameter	Description
/Partition	<p data-bbox="350 244 1083 331">This screen is available for all types of partitions, including unallocated space and extended partitions. Information on this tabbed page includes the following:</p> <ul data-bbox="366 357 1083 557" style="list-style-type: none"> <li data-bbox="366 357 1083 479">• Partition type is shown in hexadecimal followed by a text description of the partition or file system type (such as FAT, FAT32, NTFS, HPFS, and so on). The hexadecimal designation is a conventional way to display partition types.</li> <li data-bbox="366 499 1053 557">• Serial number is shown here if the partition's file system uses serial numbers. Not all file systems use serial numbers.</li> </ul> <p data-bbox="350 591 1083 644">The next section of the screen shows physical information about the partition, including the following:</p> <ul data-bbox="366 670 1083 970" style="list-style-type: none"> <li data-bbox="366 670 1083 722">• First Physical Sector shows the logical number and the location (cylinder, head, and sector) where the partition begins.</li> <li data-bbox="366 748 1083 800">• Last Physical Sector shows the logical number and the location (cylinder, head, and sector) where the partition ends.</li> <li data-bbox="366 826 1005 878">• Total Physical Sectors gives the number of sectors in the partition.</li> <li data-bbox="366 904 1083 970">• Physical Geometry shows the number of cylinders, heads, and sectors of the physical disk drive on which the partition resides.</li> </ul>

<b>Parameter</b>	<b>Description</b>
<i>/FS</i> <i>(FAT or FAT32)</i>	<p>For FAT partitions, this parameter displays a screen with the following information. The first section on this page provides the following information about the file system:</p> <ul style="list-style-type: none"> <li>• Sectors per FAT</li> <li>• Root directory capacity</li> <li>• First FAT sector</li> <li>• First Data sector</li> </ul> <p>The next section of this page gives the following information:</p> <ul style="list-style-type: none"> <li>• Number of bytes in files on the partition, the number of files, and the number of those files that are hidden</li> <li>• Number of bytes in directories on the partition, the number of directories, and the number of those directories that are hidden</li> </ul> <p>Several extensions to the FAT file system exist. The final section of this page gives the following information about FAT extensions:</p> <ul style="list-style-type: none"> <li>• Number of bytes used for OS/2 Extended Attributes and how many files and directories the Extended Attributes are associated with</li> <li>• Number of bytes used for long file names and the number of files and directories the long file names are associated with</li> </ul>
<i>/FS</i> <i>(NTFS)</i>	<p>For NTFS partitions, this parameter displays a screen with the following information. This screen shows the following file system information for the selected partition:</p> <ul style="list-style-type: none"> <li>• NTFS Version shows the version number.</li> <li>• Bytes per NTFS sector displays the number of bytes in each logical sector on the selected partition. (There are always 512 bytes in each physical sector.)</li> <li>• Cluster size</li> <li>• First MFT Cluster</li> <li>• File Record Size</li> </ul>

<b>Parameter</b>	<b>Description</b>
<i>/FS</i> <i>(NTFS</i> <i>continued)</i>	<p>The next section shows information similar to that shown by NTFS CHKDSK, including the following:</p> <ul style="list-style-type: none"> <li>• Number of files and the bytes and clusters allocated to them</li> <li>• Of the clusters used in files, the number of wasted bytes resulting from the cluster size</li> <li>• Number in indexes (directories) and the space allocated to them, shown in bytes and clusters</li> </ul> <p>Space reserved for other system structures, shown in both bytes and clusters</p>
<i>/FS</i> <i>(HPFS)</i>	<p>For HPFS partitions, this parameter displays a screen with the following information. The first section on this page provides the following information about the file system:</p> <ul style="list-style-type: none"> <li>• Partition status (that is, is partition active?)</li> <li>• DirBlock sectors</li> <li>• Free DirBlocks</li> <li>• Hot Fixes Used</li> </ul> <p>The next section of this page gives the following information:</p> <ul style="list-style-type: none"> <li>• Number of bytes in files on the partition, the number of files, and the equivalent amount of sectors</li> <li>• Number of bytes unused in file sectors</li> <li>• Number of bytes in directories on the partition, the number of directories, and the equivalent number of sectors</li> <li>• Number of bytes in file/dir Fnodes and equivalent sectors</li> <li>• Number of bytes reserved by system and equivalent sectors</li> <li>• Number of bytes in extended attributes</li> </ul>

## **Label**

### **Syntax**

Label [/GetLabel ] [/SetLabel="NAME"]

The default with no parameter is GetLabel, which reports the label of the selected partition.

The /SetLabel option lets you change the name of a selected partition. Labels can be up to 11 characters for FAT, FAT32, and HPFS partitions, 32 characters for NTFS partitions, and 16 characters for Linux Ext2 partitions. Labels for FAT, FAT32, or HPFS partition types follow the same rules as DOS names, with two exceptions: spaces are allowed, and no period is required between the first eight characters and the last three.

## **Merge**

### **Syntax**

```
Merge /Target={First | Second} /Folder="NAME" [/FS={FAT | FAT32 | NTFS}]
```

Use Merge to join two FAT or FAT32 partitions that are adjacent (unallocated space can exist between them, however) to each other on a hard disk. This is useful if you have reached the maximum number of partitions on your disk, but you do not want to delete a partition.

You can also use Merge to join two adjacent NTFS partitions with the same cluster size and version type. To ensure that two NTFS partitions have the same cluster size and version type, right-click a partition in the partition map, then select **Info** from the menu. Click the **NTFS Info** tab, and view the cluster size and version type.

For the Merge command to work correctly, two adjacent FAT or FAT32 partitions, or two NTFS partitions with the same cluster size and version type must be selected. You should not merge any partitions that include an operating system.

The Merge command must be preceded by the following commands:

- Select Disk {Num}
- Select Partition {{Number} | {PartitionLetter} | {"Volume Label"} | First | Last | Next | Previous | Extended}
- Select Merge Partition {Next | Previous}

<b>Parameter</b>	<b>Description</b>
/Target	Specifies which partition you want to keep. For example, if you choose First, the second partition you selected will merge into the first partition you selected.

<b>Parameter</b>	<b>Description</b>
------------------	--------------------

---

/Folder	Specifies the folder name for the data that is merged into the target partition. For example, if you had a DATA partition and a BACKUP partition and you were keeping the DATA partition, you could specify “BACKUP” as the folder name. After the merge, all the data from your BACKUP partition would be in a BACKUP folder inside the DATA partition.
/FS= FAT   FAT32   NTFS	<i>(Optional)</i> You can specify the resulting file system. If you do not use this parameter, PartitionMagic Pro will choose the best file system automatically.

### **Move Left**

#### **Syntax**

Move Left {Max | Min | Value }

Move a partition to the left.

<b>Parameter</b>	<b>Description</b>
------------------	--------------------

---

Max	Move the partition as far to the left as possible. Flush with the previous partition or beginning of the drive.
Min	Move the partition to the left, the minimum amount possible (1 cylinder).
Value	Move the partition left by the amount of the value specified (in megabytes).

### **Move Right**

#### **Syntax**

Move Right {Max | Min | Value }

Move a partition to the right.

<b>Parameter</b>	<b>Description</b>
------------------	--------------------

---

Max	Move the partition as far to the right as possible. Flush with the next partition or end of the drive.
-----	--

<b>Parameter</b>	<b>Description</b>
Min	Move the partition to the right the minimum amount possible (1 cylinder).
Value	Move the partition right by the amount of the value specified (in megabytes).

### ***Move Space After***

#### **Syntax**

Move Space After {Max | Value }

Similar to Move Left and Move Right.

<b>Parameter</b>	<b>Description</b>
Max	Same as Move Left Max. Makes as much space after the partition as possible by moving the partition left.
Value	Moves the partition left or right such that the space after is equal to the value specified if possible (specified in megabytes).

### ***Move Space Before***

#### **Syntax**

Move Space Before {Max | Value }

Same as Move Right.

<b>Parameter</b>	<b>Description</b>
Max	Same as Move Right Max. Makes as much space before the partition as possible by moving the partition right.
Value	Moves the partition right such that the space before is equal to the value specified if possible (specified in megabytes).

### ***Resize***

#### **Syntax**

```
Resize {Max | Min | Value } [/SetClusterSize={512 | 1 | 2 | 4  
| 8 | 16 | 32 | 64}]
```

Resize a partition.

<b>Parameter</b>	<b>Description</b>
Max	Resizes to the maximum size possible. The right edge of the partition will be flush with end of drive or next partition, if possible.
Min	Resizes to the minimum possible (determined by the size of the data).
Value	Resize to value specified (in megabytes).
/SetClusterSize	Will set the cluster size to the size specified during the resize operation. The cluster size must be valid for the partition size specified.

### ***Resize Larger***

#### **Syntax**

```
Resize Larger {Max | Min | Value} [/SetClusterSize={512 | 1  
| 2 | 4 | 8 | 16 | 32 | 64}]
```

Resize a partition larger by specifying the incremental change in size. See also, “Resize.”

<b>Parameter</b>	<b>Description</b>
Max	Same as Resize Max. Make the partition as big as possible.
Min	Grows the size of the partition by the smallest amount possible (1 cylinder).
Value	Grows the size of the partition by the size specified (in megabytes).
/SetClusterSize	Sets the cluster size to the size specified during the resize operation. (The cluster size must be valid for the partition size specified.)

### ***Resize Left Boundary***

#### **Syntax**

```
Resize Left Boundary {Max | Min | Value }
```

Resize the extended partition by moving the left boundary. **This operation is for extended partitions only.** .

Parameter	Description
Max	Resizes to the maximum size possible. The left edge of the partition will be flush with the left end of drive or previous partition, if possible.
Min	Resizes to the minimum possible (determined by the size of the data).
Value	Resize to value specified (in megabytes).

### ***Resize Left Boundary Larger***

#### **Syntax**

```
Resize Left Boundary Larger {Max | Min | Value }
```

Resize an extended partition larger by specifying the change in position of the left boundary. **This operation is for extended partitions only.**

Parameter	Description
Max	Same as Resize Left Boundary Max. Make the partition as large as possible.
Min	Increase the size of the partition by the smallest amount possible (1 cylinder).
Value	Increase the size of the partition by the size specified (in megabytes).

### ***Resize Left Boundary Smaller***

#### **Syntax**

```
Resize Left Boundary Smaller {Max | Min | Value }
```

Resize an extended partition smaller by specifying the change in position of the left (or right) boundary. **This operation is for extended partitions only.** See also, "Resize Left Boundary."

<b>Parameter</b>	<b>Description</b>
Max	Same as Resize Left Boundary Min. Make the partition as small as possible.
Min	Partition's size will be decreased by the minimum amount possible (1 cylinder) by moving the right boundary.
Value	Partition's size will be decreased by the amount specified (in megabytes).

### ***Resize Smaller***

#### **Syntax**

```
Resize Smaller {Max | Min | Value}    [/SetClusterSize={512 | 1 | 2 | 4 | 8 | 16 | 32 | 64 }]
```

Resize a partition smaller by specifying the incremental change in size. See also, "Resize."

<b>Parameter</b>	<b>Description</b>
Max	Same as Resize Min. The partition will be as small as possible.
Min	Partition's size will be decreased by the minimum amount possible (1 cylinder).
Value	Partition's size will be decrease by the amount specified (in megabytes).
/SetClusterSize	Sets the cluster size to the size specified during the resize operation. The cluster size must be valid for the partition size specified.

### ***Resize Space After***

#### **Syntax**

```
Resize Space After {Max | Min | Value}
[/SetClusterSize={512 | 1 | 2 | 4 | 8 | 16 | 32 | 64 }]
```

Resize a partition by specifying the unallocated space desired after the partition after the resize is completed. This operation is for all partitions, including extended partitions. See also, "Resize."

<b>Parameter</b>	<b>Description</b>
Max	Resizes so that the space after the partition is as large as possible. The partition is as small as possible.
Min	Resizes so that the space after the partition is as small as possible. The partition is as large as possible.
Value	Sizes the partition such that the space after is the size of value (in megabytes).
/SetClusterSize	Sets the cluster size to the size specified during the resize operation. Cluster size must be valid for the partition size specified.

### ***Resize Space Before***

#### **Syntax**

```
Resize Space Before {Max | Min | Value }
```

Resize an extended partition by specifying the unallocated space desired before the partition after the resize is completed. **This operation is for extended partitions only.**

<b>Parameter</b>	<b>Description</b>
Max	Resizes so that the space before the partition is as large as possible. The partition is as small as possible.
Min	Resizes so that the space before the partition is as small as possible. The partition is as large as possible.
Value	Sizes the partition such that the space before is the size of value (in megabytes).

### ***Resize Root***

#### **Syntax**

```
Resize Root {Value | Min | Max }
```

Changes the number of entries in the root directory of a FAT partition.

<b>Parameter</b>	<b>Description</b>
Value	Must be a value between 512 - 1024. This will change the maximum number of root entries possible for this partition. The number actually set will be the closest number possible to the number specified.
Min	Sets the partition to having the smallest possible maximum number of root entries.
Max	Sets the maximum number of root entries for the partition to the largest number possible.

### **Select**

You do not need to select a disk when selecting a partition by letter or volume. All the disks are searched in order until the specified letter or volume name is encountered. This means that if two partitions have the same volume name, the first partition found is the one that is selected.

The Next and Previous commands are relative to a previous selection of the same type. You must already have a partition selected before using Select Partition Next or Select Partition Previous. Likewise, you must have an unallocated space selected before using Select Unallocated Next or Select Unallocated Previous.

### **Select Disk**

#### **Syntax**

```
Select Disk {Number}
```

Use to select a hard disk.

There are no parameters.

### **Select Partition**

#### **Syntax**

```
Select Partition [{Number} | {Letter} | {"Volume"} | First | Last | Next | Previous | Extended]
```

Select a partition. If you select a partition by letter, you must type the letter in uppercase.

There are no parameters.

### **Select Unallocated**

#### **Syntax**

```
Select Unallocated [{Number} | First | Last | Largest | Next  
| Previous | After Selected Partition | Before Selected  
Partition}]
```

Select unallocated space on the disk.

There are no parameters.

### **Select Merge Partition**

#### **Syntax**

```
Select Merge Partition {Next | Previous}
```

Select a partition to merge with an adjacent partition. This command must be preceded by a Select Partition command.

There are no parameters.

### **Select Destination Disk {Number}**

Select the destination hard disk. This command is used with the Copy command.

There are no parameters.

### **Select Destination Unallocated**

#### **Syntax**

```
Select Destination Unallocated | {{Number} | First | Last |  
Largest | After Partition {Number} | Before Partition  
{Number} | Next | Previous}
```

Select the destination unallocated space on the partition. This command is used with the Copy command.

There are no parameters.

### **Set Active**

Mark the selected partition as the active, or bootable, partition.

There are no parameters.

### ***Set Allow User Cancel {ON | OFF}***

#### **Syntax**

```
Set Allow User Cancel [ON | OFF]
```

Set allow user cancel OFF to prevent interrupting a script that is running. This command is turned on by default. If the cancel button is grayed out, then it is turned off.

Use this command when you play a script and you do not want the script to be interrupted while it is running. Interrupting a playing script could potentially damage your hard drive.

### ***Set Default Bad Sector Test State { ON | OFF}***

Set the bad sector testing ON or OFF for all partitions on the currently selected drive. The "/BadSectorTest" option overrides this setting.

### ***Set Drive Read Only Mode { ON | OFF}***

Set the read-only flag ON or OFF for all partitions on the currently selected drive. When set on for a drive, modifications to the partitions on that drive will not be allowed. Some changes to boot.ini files may be allowed if they exist on the read-only drive when add, delete or copy partition operations are done.

### ***Set Ignore OS/2 EA Errors { ON | OFF }***

If you do not have OS/2 on your system, you can turn this option ON to ignore OS/2 Extended Attribute errors when it checks a FAT partition.

### ***Set NT 64K FAT Clusters { ON | OFF }***

If this option is set ON, you can create FAT partitions with 64 K clusters, which allows you to use PartitionMagic Pro to create FAT partitions up to 4 GB.

You should only use 64 K clusters with Windows NT or Windows 2000. If you use other operating systems, you should not use 64 K clusters.

### ***Set Force User Logoff { ON | OFF }***

Set user logoff to on or off. When set to on, it will force all users to log off the system just before the remainder of the script is played.

## **Show**

### **Syntax**

Show {Partitions | Preference | Destination}

Display a summary of information for the selected parameter.

<b>Parameter</b>	<b>Description</b>
Partition	Displays information about partitions. You must select the disk.
Preference	Displays the preferences from the <b>General ► Preferences</b> menu.
Destination	Displays information about the destination disk. The command must be preceded by Select Destination Disk command.

## **Unhide**

Unhide the currently selected partition. See also, “Hide.”

There are no parameters.

## **Script File Statements**

You can use script statements to modify the flow of a script based on conditions or variables. You can insert a statement using ScriptBuilder by clicking **Insert ► Statement**.

### **Variables**

Create a variable with a name you specify to assign a value for later use in conditional or mathematical expressions.

### **Syntax**

```
Dim {Put_variable_name_here}
```

For example, the following lines create a variable named “Number” and assign it the value of 2.

```
Dim Number  
Number = 2
```

## ***If...End If***

Perform different statements depending on the specified conditions. Each If { } Then statement must be followed by an End If statement. The Else If { } Then and Else statements are optional.

### **Syntax**

```
If {Put_condition_here} Then
// Put commands here
Else If {Put_condition_here} Then
// Put commands here
Else
// Put commands here
End If
```

## ***Do...Loop While***

Perform repeated statements until the condition is false or zero. The statements between the Do and the Loop While { } commands are always played once before the condition is tested.

### **Syntax**

```
Do
// Put commands here
Loop While {Put_condition_here}
```

## ***Do While...Loop***

Perform repeated statements until the condition is false or zero. The statements between the Do While { } and the Loop commands are never played if the condition is false or zero the first time it is tested.

### **Syntax**

```
Do While {Put_condition_here}
// Put commands here
Loop
```

## Exit Loop

Use the Exit Loop command to jump out of a Do...Loop While or a Do While...Loop statement before reaching the end of the loop.

At least one command must exist after the loop or the script will stop with an error.

## Operators

PartitionMagic supports the following operators in scripts.

Operator	Action
<b>Conditional</b>	
<	Returns true if the left operand is less than the right operand.
=	Returns true if the operands are equal.
>	Returns true if the left operand is greater than the right operand.
<=	Returns true if the left operand is less than or equal to the right operand.
>= or =>	Returns true if the left operand is greater than or equal to the right operand.
<>	Returns true if the two operands are not equal.
<b>Mathematical</b>	
+	Adds the left and right operands.
-	Subtracts the right operand from the left operand.
*	Multiplies the left and right operands.
/	Divides the left operand by the right operand.
<b>Assignment</b>	
=	Assigns the value of the right operand to the left operand. For example, X=3 means that X is set to the value of 3. You can also use the operator inside parameters.

Numbers in scripts can be treated as true or false. Any statement that allows a condition (the IF and both loops), you can enter a numeric expression (for example, queries and arithmetic). PartitionMagic Pro scripting treats a zero result as FALSE and a non-zero result as TRUE. For example, the following two lines both evaluate as TRUE or FALSE and have the same effect and result:

```
IF GetTotalPartitions then ...  
IF GetTotalPartitions > 0 then ...
```

## **Operator Examples**

The following example uses a variety of operators throughout the script.

```
// PowerQuest PartitionMagic Pro Script File  
Select Disk 2  
Select Partition D  
//Resize double  
Resize Larger GetSelectedPartitionSize * 2  
//Resize to half the unused space  
Resize Smaller GetUsedAmount / 2  
// Create a partition using half the unallocated space  
Select Unallocated Largest  
Create /FS=FAT32 /Size=GetUnallocatedSize * 0.5  
// Create a 300 MB partition if there is room  
Dim Num  
Num = 300  
Select Unallocated Largest  
If GetUnallocatedSize > Num Then  
    Create /FS=FAT32 /Size=Num  
End If
```

## **Queries**

To insert a query in a script using ScriptBuilder, click **Insert ► Query**. Queries are inserted into the script edit pane at the location of your cursor and replace any selected text.

You can assign values returned from queries to variables or you can use them in conditional or mathematical expressions. The following table shows the queries that are available for your use.

<b>Query</b>	<b>Use and Syntax</b>
Number of Disks	Get the total number of physical disks on the selected system. Syntax: <code>GetTotalDisks</code>
Number of Partitions	Gets the total number of partitions on the selected disk. Syntax: <code>GetTotalPartitions</code>
Number of Unallocated.	Gets the total number of unallocated spaces on the selected disk. Syntax: <code>GetTotalUnallocatedSpaces</code>
Disk Size	Get the size of the currently selected disk, in megabytes. Syntax: <code>GetDiskSize</code>
Disk Allocated {Size   Percentage}	Get the size of disk allocated to partitions, in megabytes or as a percentage. Syntax: <code>GetAllocatedSize</code> <code>GetAllocatedPercent</code>
Disk Unallocated {Size   Percentage}	Get the size of remaining unallocated space on the currently selected disk, in megabytes or as a percentage. Syntax: <code>GetUnallocatedSize</code> <code>GetUnallocatedPercent</code>
Partition Size	Get the size of the partition, in megabytes. Syntax: <code>GetSelectedPartitionSize</code>

Query	Use and Syntax
Partition Number	<p>Get the partition number of the selected partition. You can then use the returned value with the Select Partition {Number} command. A return of "0" indicates that the partition does not exist. (For example, the GetPartitionNumber Extended command returns "0" if there is no extended partition.)</p> <p>Syntax: <code>GetPartitionNumber {&lt;number&gt;   &lt;Letter&gt;   "NAME"   First   Last   Next   Previous   Extended }</code></p>
Partition Used	<p>Get the size of used space on a partition, in megabytes or as a percentage.</p> <p>Syntax: <code>GetUsedAmount</code> <code>GetUsedPercent</code></p>
Partition Unused	<p>Get the size of unused space on a partition, in megabytes or as a percentage.</p> <p>Syntax: <code>GetUnusedAmount</code> <code>GetUnusedPercent</code></p>
Type	<p>Determine if the file system of the current partition is the type specified.</p> <p>Syntax: <code>{IsFAT   IsFAT32   IsNTFS   IsHPFS   IsLinuxExt2   IsLinuxSwap   IsExtended   IsUnallocated   IsUnformatted}</code></p>
Status	<p>Determine if the status of the current partition is active or hidden.</p> <p>Syntax: <code>{IsActive   IsHidden}</code></p>
Primary/Logical	<p>Determine if the class of the current partition is primary or logical.</p> <p>Syntax: <code>{IsPrimary   IsLogical}</code></p>
Unallocated Size	<p>Get the size of the unallocated space, in megabytes.</p> <p>Syntax: <code>GetSelectedUnallocatedSize</code></p>

## Query

## Use and Syntax

---

Unallocated Number	Get the space number of the specified unallocated space. You can then use the returned value with the Select Partition {Number} command. A return of 0 (zero) indicates that the unallocated space does not exist. (For example, the GetUnallocatedNumber Next command returns 0 if there are no unallocated spaces following the last selection.)
--------------------	--

Syntax: GetUnallocatedNumber {<Number> | First | Last | Largest | Next | Previous | After Selected Partition | Before Selected Partition}

### **Examples**

For example, in the following code, the variable dNumParts is defined, then assigned the total number of partitions on disk 1.

```
Dim dNumParts
Select Disk 1
dNumParts = GetTotalPartitions
```

In the following code example, Partition C: is selected. If the partition is FAT, it is converted to FAT32.

```
Select Partition C
If IsFAT Then
Convert to FAT32
End If
```

In the following example, the largest unallocated space on disk 1 is selected. If it is larger than 1000 MB, PartitionMagic Pro creates two partitions, each using approximately half of the unallocated space.

```
Select Disk 1
Select Unallocated Largest
If GetSelectedUnallocatedSize > 1000 Then
Create /FS=FAT /Size = GetSelectedUnallocatedSize / 2
Select Unallocated After Selected Partition
Create /FS=FAT
End If
```

## Script Suggestions and Notes

Although it is not necessary, PowerQuest recommends that you check each of the partitions that will be modified at the beginning of the script. Because a script file will terminate as soon as an error occurs, checking each of the partitions first will keep the script from making any changes before it finds errors.

Partitions must start on cylinder boundaries. For example, if you specify 10 MB, the real value could be 10.2 MB. The difference between the specified and actual values varies depending on the geometry of the drive.

When specifying an amount for one of the script options, the program will allow a margin of error of 1 cylinder above or below that amount (or a range of 2 cylinders centered on the amount specified). For example, if 10 MB were specified for a resize and a cylinder was .5 MB, that the operation would complete successfully if it could resize the partition to at least 9.5 MB. The actual range would be 9.5 to 10.5 MB. If it could not resize the partition within this range, it will return an error.

Under normal operation, if the script determines that it will not be able to reboot the machine after making the changes specified in the script, the script will terminate with an error. This condition will occur under OS/2 if the DOS.SYS file is not in the CONFIG.SYS (such as when you boot from the utility disks). You should include ALLOW MANUAL REBOOT as the first script statement in the script if performing a manual reboot from the keyboard is not a problem.

All commands must be contained within one line. They cannot start on one line and finish on the next. The maximum length of a script line is 180 characters, which should be sufficient for any command.

Use extreme caution when selecting a partition by its number. The select by number feature must be available to select unallocated space or partitions that have no drive letter or label. The problem with selecting a partition by number is that the numbers can change throughout a script. If you select partition 2 and move it to the right, any unallocated space that has been moved from the right to the left side of the partition will now become partition 2. (The partition moved will still be selected regardless of the number). Using the Select Unallocated After Selected Partition and Select Unallocated Before Selected Partition commands is usually preferable to selecting unallocated space by partition number. With most operations, the partition selected after an operation will be the partition operated on. For example, Resize and Move will always leave the partition operated on as the selected partition after the operation. On a Create command, the partition created will be selected after the command whether it is at the beginning or end of the unallocated space it was created in. If there is any doubt as to which partition will be selected after an operation, you can use the DOS (rescue disk) version of the program (without scripting)

and perform the same operation on a test machine and observe which partition is selected after the operation. You can also use the Show Partitions command to show the current status of partitions.

## Sample Scripts

### **Scenario 1: General Example**

You have primary C:, D:, and E: drives. There is no unallocated space on the disk. You want to take 10 MB from D: and add it to E:.

```
SCRIPT1.PQS
//Check the partitions to be operated on first
Select Partition D
Check
Select Partition E
Check
//Select the first partition I want to change
Select Partition D
//Shrink the partition by 10 megabytes
Resize Smaller 10
//Select the partition to add the 10 meg to
Select Partition E
//Move the partition as far as possible to the left so that
//the unallocated space just created will be on the right
//edge (the end)
Move Left Max
//Take up all of the available space
Resize Larger Max
```

### **Scenario 2: General Example**

You have one large C: partition on the drive. The drive is 1.2 GB in size. You have only 300 MB of data on the partition and would like to create logical drives D: and E:. The E: drive needs to be 300 MB, and the rest of the disk space is to be split between the C: and D: drives.

```
SCRIPT2.PQS
```

```

//Check the partition first
Select Partition C
Check
//Partition C is already selected so shrink it to 450 MB
Resize 450
//Since the C partition is still selected after the resize,
//we need to select the unallocated space created
//after C.
Select Unallocated After Selected Partition
//Create the extended partition to the default size, which
//will be all of the unallocated space currently selected
Create /FS=EXTENDED
//The extended partition is now selected, and we want to
//select the next unallocated space in the extended
//partition.
Select Unallocated After Selected Partition
//Create the partition that we need to be 300 MB first at the
//end of the unallocated space that is currently selected.
//(what will be the E partition)
Create /FS=FAT /Label="DBFILES" /Size=300 /Position=END
//Select the rest of the unallocated space within the
//extended partition. Since the last partition was created
//at the end of the unallocated space, we need to move to
//the unallocated space previous to the selected partition
Select Unallocated Before Selected Partition
//Create the partition in the rest of the unallocated space
Create /FS=FAT /Label="APPS"

```

### **Scenario 3: General Example**

You have C:, D: and E: partitions on one physical drive. There is no unallocated space on the disk.

The C: partition is a primary partition, and the D: and E: partitions are logical drives in an extended partition. You want to create an F: partition with 40 MB of unused space that is in the C: partition. The F: partition will be a FAT partition with a volume label of "DATA," Since the drive is fairly new, you would like to skip bad sector testing for all operations.

```

SCRIPT3.PQS
//Check all of the partitions first
Select Partition C
Check
Select Partition D
Check
Select Partition E
Check
//Since a partition on this drive had already been selected,
//we can set the default bad sector testing to off for this
//drive
Set Default Bad Sector Test State Off
//Select the C partition and shrink it by 40 MB
Select Partition C
Resize Smaller 40
//Select the extended partition and resize the left
//boundary to the right edge of the C partition (max),
//putting the unallocated space within the extended
//partition. To select an extended partition, the drive must
//first be selected, and then the partition.
Select Disk 1
Select Partition Extended
Resize Left Boundary Max
//Select the D partition and move it to the left, essentially
// flush against the Extended and C partitions, leaving the
// unallocated space between the D and E partitions
Select Partition D
Move Left Max
//Select the E partition and move it as far as possible to
//the left, so that the unallocated space will be at the end
//of E, within the extended partition
Select Partition E
Move Left Max
//The unallocated space is now after E and the user
// can create an F partition (logical drive)

```

```
//Move to the unallocated space after E
Select Unallocated After Selected Partition
//Create the FAT partition called DATA with all defaults.
//This will use all of the size available in the unallocated
//space.
Create /FS=FAT /Label="DATA"
```

#### **Scenario 4: General Example**

The user has a C partition which is a primary having 100 MB. The next partition is a hidden partition called NT\_OS which is FAT and a primary partition at 100 MB. There are also 2 logical drives, D and E in an extended partition which are each 70 MB.

The user wants to reduce both the C and Hidden partitions to 60 MB, add 40 MB to the D partition, and create an F partition (NTFS) with the remaining unallocated space. The user also wants to convert the hidden partition from FAT to NTFS. The user also does not care whether the machine can reboot under program control or not.

```
SCRIPT4.PQS
//Inform that a manual boot is acceptable
Allow Manual Reboot
//Check all of the partitions first
Select Partition C
Check
//The hidden partition is selected by using the volume label
//in quotes
Select Partition "NT_OS"
Check
Select Partition D
Check
Select Partition E
Check
//Select the C Partition and resize it to 60 MB
Select Partition C
Resize 60
//Select the hidden partition
Select Partition "NT_OS"
```

```

//Move the partition flush against the C partition (since it
//was just resized) putting the newly created unallocated
//space after the hidden partition
Move Left Max
//Resize the hidden partition to 60 MB
Resize 60
//Convert the partition from FAT to NTFS
Convert To NTFS
//Expand the extended partition so that the unallocated
//space is now inside the expanded partition
Select Disk 1
Select Partition Extended
Resize Left Boundary Max
//Move the D partition flush against the hidden and extended
//partitions
Select Partition D
Move Left Max
//Add 40 MB to the D partition
Resize Larger 40
//Move the E partition next to the D partition
Select Partition E
Move Left Max
//The unallocated space is now available at the end of the
//extended partition so that you can create an F logical
//drive. Select the unallocated space.
Select Unallocated After Selected Partition
//Create the NTFS partition.
Create /FS=NTFS

```

### ***Scenario 5: Cluster Analyzer***

Imagine you have a 3.2 GB drive. You have a 2 MB primary partition and a 1 GB primary FAT C: partition. You also have a hidden primary FAT partition that is 1 GB. You boot multiple operating systems, and the third partition holds another operating system. You also have an extended partition with logical drives that use up the rest of the drive space.

You would like to analyze the two FAT partitions to see if you can reduce the cluster waste.

```
SCRIPT5A.PQS
// Show Cluster Waste for Partition 2
Select Disk 1
Select Partition 2
Cluster Analyzer /ShowClusterWaste
// Show Cluster Waste for Partition 3
Select Disk 1
Select Partition 3
Cluster Analyzer /ShowClusterWaste
```

You can now know you can reduce cluster waste, so you use the Cluster Analyzer to reduce waste again. You will set the third partition to the recommended size and the second partition to a cluster size of 8K.

```
SCRIPT5B.PQS
// Set Partition 3 to Recommended Cluster Size
Select Disk 1
Select Partition 3
Cluster Analyzer /SetToRecommended
// Set Partition 2 to 8K Clusters
Select Disk 1
Select Partition 2
Cluster Analyzer /ClusterSize=8
```

### **Scenario 6: Copy**

You just installed a new drive. You would like to copy the first three partitions on drive 1 to drive 2. Drive 2 is formatted and is unused space.

```
SCRIPT6.PQS
// Select Disk 1, then select and check partitions 3, 2, and
//1.
Select Disk 1
Select Partition 3
Check
```

```

Select Partition 2
Check
Select Partition 1
Check
// Select Destination Disk 2, Copy Partition 1
Select Destination Disk 2
Select Destination Unallocated First
// Copy First Partition
Copy
// Select Disk 1, Partition 2
Select Disk 1
Select Partition 2
// Select Destination Disk 2, Copy Partition 2
Select Destination Unallocated First
// Copy Second Partition
Copy
// Select Disk 1, Partition 3
Select Disk 1
Select Partition 3
// Select Destination Disk 2, Copy Partition 3
Select Destination Unallocated First
// Copy Third Partition
Copy

```

### ***Scenario 7: Info***

You have a FAT C: partition that you would like to get some information about.

You do not know how big the partition is or much about it. You would like to know about the disk usage, the cluster waste, partition information, and file system information.

```

SCRIPT7A.PQS
Select Disk 1
Select Partition 1
Info /Usage /Waste /Partition /FS

```

This could also be accomplished as follows.

```
SCRIPT7B.PQS
Select Disk 1
Select Partition 1
Info /Usage
Info /Waste
Info /Partition
Info /FS
```

### ***Scenario 8: Adding Unused Space on Disk to Primary Partition***

Disk 1 contains a primary partition and an unknown number of logical partitions. This script takes 10 MB of unused space (if available) from each logical partition and adds the unused space to the primary partition.

```
SCRIPT8.PQS
// Select the last partition
Select Disk 1
Select Partition Last
// Loop thru the logical partitions and remove 10 MB of
//unused space (if available).
Do While IsLogical
    If GetUnusedAmount > 10 Then
        Resize Smaller 10
        Move Right Max
    End If
    Select Partition Previous
Loop
// Resize the extended partition to fit the logical
// partitions
Select Partition Extended
Resize Left Boundary Min
// Expand the primary partition to include the unused space
Select Partition Previous
Resize Larger Max
```