

PathMan is a command-line tool that enables you to add or remove components of both the system and user paths. It can modify any number of paths in a single call and includes error checking that can handle path abnormalities such as repeated entries, adjacent semicolons, and missing entries.

Microsoft® Windows® 2000 manages two different types of paths; the current user's path and the system path. Windows 2000 keeps a separate user path for every user on a system, but there is only one global system path. When a user logs on to a Windows 2000 computer, these two paths are combined to create the user's full path. For backward compatibility, Windows 2000 also loads path information from old Autoexec.bat files left in the root directory.

A non-administrative user can modify his or her own user path. But in order to modify the system path, the user of PathMan must have Administrator privileges.

PathMan does not manage information contained in old Autoexec.bat files.

PathMan also broadcasts a message to all top-level windows notifying them that the environment has changed. This causes many applications (like the system shell) to update their environments, thereby obtaining the modified path.

PathMan is included in the *Microsoft® Windows® 2000 Resource Kit*.

## Contents

PathMan Syntax	1
PathMan Examples	2
Checking for Path Abnormalities	2
The PathMan Return Value	3

## PathMan Syntax

PathMan accepts four command line arguments:

`/as path[;path[;path ...]]`

Adds the semicolon-separated paths to the system path

`/au path[;path[;path ...]]`

Adds the semicolon-separated paths to the user path

`/rs path[;path[;path ...]]`

Removes the semicolon-separated paths from the system path

`/ru path[;path[;path ...]]`

Removes the semicolon-separated paths from the user path

The two add flags, **/as** and **/au**, will not add a path that already exists. The two remove flags, **/rs** and **/ru**, will do nothing if the path does not currently exist.

PathMan can accept multiple arguments in a single call, each identified by one of the four command line flags (**/as**, **/au**, **/rs**, **/ru**). Each flag can be added in any order as many times as desired. For each argument, PathMan can accept any number of paths to modify (represented above by *path*). Each path must be separated by a semicolon.

Once the path modifications have been made, PathMan broadcasts a message to all top-level windows notifying them that the environment has changed. This causes many applications (including the system shell) to update their environments, thereby obtaining the modified path.

## PathMan Examples

The following examples show how PathMan uses command line flags.

### Example 1

```
pathman /au c:\temp;c:\users\name;d:\utils
```

In this example, the **/au** flag tells PathMan that the next set of semicolon separated paths should be added to the current user's path. All three paths (c:\temp, c:\users\name, and d:\utils) will be added to the user's path if they do not already exist.

### Example 2

```
pathman /au c:\temp;c:\users\name;d:\utils /ru d:\data /as d:\data;c:\reskit
```

Aside from the additions to the current user's path, d:\data is removed from the current user's path and both d:\data and c:\reskit are added to the system path. PathMan can accept any number of path modifications this way.

### Example 3

```
pathman /au c:\temp;c:\app\bin /rs %systemroot% /au c:\app2\bin
```

This command adds (c:\temp, c:\app\bin, and c:\app2\bin) to the user path and removes (%SystemRoot%) from the system path.

## Checking for Path Abnormalities

PathMan can handle many path abnormalities. These include:

- Additional leading semicolons
- Additional trailing semicolons
- Multiple consecutive semicolons
- Double path additions using **/au** or **/as**
- Double path removal using **/ru** or **/rs**
- Addition and removal of the same path (results in removal)
- Addition of an already existing paths
- Removal of a non-existing paths

PathMan does not check the validity of new paths, however.

## The PathMan Return Value

The PathMan return value is complicated, because the tool needs to return a lot of information in a single 32-bit value. In order to return status information for each of the four possible flags (**/as** **/au** **/rs** **/ru**) it accepts as arguments, PathMan treats the return value as four separate 8-bit bytes instead of one 32-bit integer. Each of the four bytes in the return value indicates what action was taken with respect to

each of the four flags accepted by PathMan. The return value (*wwxyzz*) is broken down as follows:

System path additions.

User path additions.

System path removals.

User path removals.

Each byte in the return code can have a value of:

No changes were made.

At least one change was made.

An error occurred. No changes were made.

In Example 1:

```
pathman /au c:\temp;c:\users\name;d:\utils
```

If we assume that there were no errors and at least one of the paths was added to the user's path, we get a return value of (in hexadecimal followed by decimal in parentheses):

```
0x00010000 (65536)
```

In Example 2:

```
pathman /au c:\temp;c:\users\name;d:\utils /ru d:\data /as d:\data;c:\reskit
```

If we again assume that everything worked, we get a return value of:

```
0x01010001 ( 16842753)
```

If the user did not have Administrator privileges, the return value would have been:

```
0x02010001 ( 33619969)
```

If PathMan is called with incorrect parameters, the return value is all errors:

```
0x02020202 (33686018)
```

## Feedback

For questions or feedback concerning this tool, please contact [rkinput@microsoft.com](mailto:rkinput@microsoft.com).

© 1985-2000 Microsoft Corporation. All rights reserved.