

---

Subject: Re: File/Folder deleting, Recycle Bin and Read only files

Posted by [mirek](#) on Wed, 06 Jan 2010 06:49:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

koldo wrote on Tue, 05 January 2010 04:02Hello all

Presently we have this basic functions to delete files and folders:

```
bool FileDelete(const char *filename)
```

```
bool DirectoryDelete(const char *dirname)
```

They delete forever a file/folder if it is not read-only (and of course if it is available).

It would be great if they would manage as transparently as possible the Recycle Bin and the Read-Only files.

Some possible options would be:

1.- To add to this functions new parameters. For example:

```
bool FileDelete(const char *filename, bool recycleBin = false, bool delReadOnly = false)
```

```
bool DirectoryDelete(const char *dirname, bool recycleBin = false, bool delReadOnly = false)
```

or

```
enum DeleteFlags {DEL_FOREVER = 0, DEL_RECYCLE = 1, NOT_DEL_READONLY = 0, DEL_READONLY = 2};
```

```
bool FileDelete(const char *filename, int flags = 0)
```

```
bool DirectoryDelete(const char *dirname, int flags = 0)
```

2.- To add functions to handle this:

```
void SetDeleteToRecycleBin(bool val = false);
```

```
void SetDeleteReadOnly(bool val = false);
```

or

```
enum DeleteFlags {DEL_FOREVER = 0, DEL_RECYCLE = 1, NOT_DEL_READONLY = 0, DEL_READONLY = 2};
```

```
void SetDeleteFlags(bool val = false);
```

What do you think about it ?

Best regards

Koldo

I do not think doing it this way is a good idea.

Path.h functions are meant to be base level API encapsulation. At these levels, OS has no notion about recycle bin.

Recycle bin, OTOH, is a high level, basically OS shell function.

Mirek