Delete Library API Manual

Last edited at 2015.06.12

1. Delete Method

Delete Library supports delete algorithm including:

DEL\_METHOD\_00, // erase file with all digit 0

DEL\_METHOD\_FF, // erase file with all digit 1

DEL\_METHOD\_RANDOM, // erase file with random value

DEL\_METHOD\_NSA, // erase file with NSA Standard

DEL\_METHOD\_OLD\_NSA, // erase file with old NSA Standard

DEL\_METHOD\_DOD\_STD, // erase file with DOD Standard

DEL\_METHOD\_NATO, // erase file with NATO Standard

DEL\_METHOD\_GUTMANN // erase file with gutmann algorithm

1. Delete Operation Result

API return value defined as below:

DEL\_RESULT\_OK, // Operation executes OK.

DEL\_RESULT\_OPERATION\_CANNOT\_BE\_EXECUTED,

// Operation cannot be executed.

DEL\_RESULT\_INVALID\_FILE\_PATH,// Inputing file path is invalid

DEL\_RESULT\_INVALID\_PARAMETER,// Inputing parameter is invalid

DEL\_RESULT\_UNKNOWN\_ERROR // Unknown unexpected error occurred

1. class DeleteLibrary

Delete library exports the class “DeleteLibrary” to implement all functions

1. DeleteOperationResult SetDeleteMethod(DeleteMethod method);

Function: set delete method, if deleting is in progress, method cannot be set.

DeleteLibrary will delete files using input method.

Parameter: method, defined as DeleteMethod

Return: DeleteOperationResult

1. DeleteMethod GetDeleteMethod();

Function: get delete method using by DeleteLibrary.

Return: DeleteMethod

1. DeleteOperationResult SetLogPath(const TCHAR\* logPath);

Function: set log path. If deleting is in progress, log path cannot be set.

DeleteLibrary will write log to inputing logPath when file deleted successfully or failed.

Log format is "Delete (file path), User:(user name), Method:(method), Time:(execution time), Result:( Success or Failure)"

If log path is not be set or set with empty string, log will not be output

Parameter: logPath, the path of log file. logPath will be deep copied immediately after entering function.

Return: DeleteOperationResult

1. const TCHAR\* GetLogPath();

Function: get current using log file path

Return: log path. The life span of the returned const TCHAR\* lasts until when the next time SetLogPath function is called or DeleteLibrary is destoryed.

1. ULARGE\_INTEGER GetTotalSizeOfProcessing();

Function: get total size of all file to delete.

NOTE: This value is timed by erase times of each delete method

For example, delete a file of size 1000 with method NSA, the value = 1000 \* 3 = 3000.

Erase times of each delete method as below:

// DEL\_METHOD\_00 = 1

// DEL\_METHOD\_FF = 1

// DEL\_METHOD\_RANDOM = 1

// DEL\_METHOD\_NSA = 3

// DEL\_METHOD\_OLD\_NSA = 4

// DEL\_METHOD\_DOD\_STD = 5

// DEL\_METHOD\_NATO = 7

// DEL\_METHOD\_GUTMANN = 35

Return: total size of all file timed by erase times.

1. ULARGE\_INTEGER GetSizeOfProcessed();

Function: get total size of all file already deleted.

NOTE: This value is timed by erase times of each delete method

Return: total size of already deleted file timed by erase times.

1. bool IsDeletingInProgress();

Function: check is deleting in progress

Return: is deleting in progress or not.

1. DeleteOperationResult StartDeletingTask(const TCHAR\* pDeletePathArray, int arrayCount);

Function: start deleting all files and folders. This function is none-blocked, deleting operation will be executed in a daemon thread.

Parameter: "pDeletePathArray" may include several files' and folders' path, no matter duplicated or overlapped.

"arrayCount" indicates how many path element in pDeletePathArray.

If arrayCount equals to 1, pDeletePathArray will be treated as a normal TCHAR\* path which ends with '\0'. If arrayCount is larger than 1, the count of TCHAR in pDeletePathArray is supposed to be MAX\_PATH \* arrayCount. In that case, every MAX\_PATH TCHAR contains a file or folder path which ends with '\0'.

“pDeletePathArray” will be deep copied immediately after entering function.

Return: DeleteOperationResult

1. DeleteOperationResult StartDeletingRecycledBin();

Function: start deleting all files and folders in RecycledBin. This function is none-blocked, deleting operation will be executed in a daemon thread.

user application should call windows API SHEmptyRecycleBin to make RecycleBin update its status after deleting finished.

Return: DeleteOperationResult

1. void StopDeletingTask();

Function: stop current deleting task

1. static bool CheckIsNormalPath(const TCHAR\* path);

Function: check input file path is a normal path

The normal path is path not a "C:\" or "windows directory" or "program files" or "program files(x86)" or other users' directory or current application's directory.

Deleting a none-normal path may cause critical error

Parameter: path, the path to check whether a normal path or not.

“path” will be deep copied immediately after entering function.

Return: bool value indicates “path” is a normal path or not.