



[首页](#) [文章](#) [用户](#) [帮助](#)

[桌面开发](#) >> [文件和文件夹](#)

VC中实现历史记录的全面清除

来源: [来自网络](#)

作者: [陈亮](#)

C++, Windows, 程序员, 初学者

发表日期: **2009**年**6**月**2**日

浏览量: **783**

收藏次数: **0**次

清除历史记录的文章很多，但是一般都不够全面。作者经过整理和收集一些资料，加以融合。得到了一个较全面的历史记录清除方法。

网站公告



[HelloCpp正式上线服务](#)



[HelloCpp招聘网站管理员](#)

搜索:

文章



打印



报告



收藏



评论



联系作者



分享

1 2 3 4 5

文章分类

[桌面开发](#)

[Web开发](#)

[企业级系统](#)

[图象/音频/多媒体](#)

[数据库](#)

[平台/框架/库](#)

[编程语言](#)

[通用编程技术](#)

[编程相关](#)

[项目管理](#)

[非技术文章](#)

[游戏开发](#)



[ClearHistory.zip - 16.18 KB\(已被下载207次\)](#)

前言

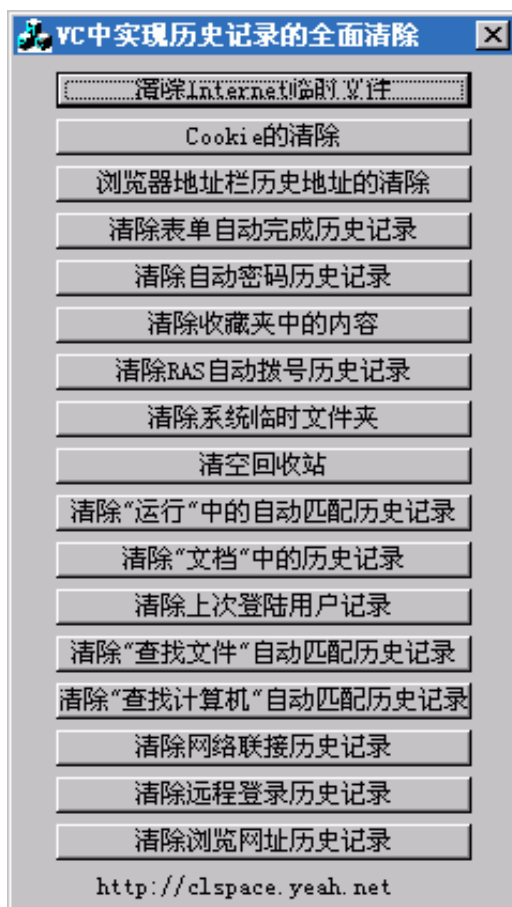
在本人用完计算机之后，往往不希望别人发现你做过什么，和浏览过什么网址。（特别是和多人共用一台机器时）。当然，你可以手工清理那些记录。但那是多么繁杂的一件事啊。我想没人愿意那样做。于是，我想用到一个软件来一次性完成上述工作。

清除历史记录的文章很多，但是一般都不够全面。本人经过整理和收集一些资料，加以融合。得到了一个较全面的历史记录清除方法。

友情连接



[北京合歌科技有限公司](#)



上图便是本文例子程序的运行效果图，我们可以从中看到该例子具体能实现的清除内容。

实现方法

首先: 在 `stdafx.h` 中要加入: `#include "atlbase.h"`。设置->Link 要引入库: `Wininet.lib` `shlwapi.lib`。在对话框的头文件中加入:

```
#include "wininet.h"
#include "UrlHist.h"

#define SWEEP_BUFFER_SIZE 10000

public:

enum DEL_CACHE_TYPE //要删除的类型。
{
File, //表示internet临时文件
Cookie //表示Cookie
};
```

以下为一些辅助函数(辅助函数的具体实现参见源代码中.限于篇幅不再给出)

```
BOOL DeleteUrlCache(DEL_CACHE_TYPE type);
BOOL EmptyDirectory(LPCTSTR szPath, BOOL bDeleteDesktopIni = FALSE, E
BOOL IsWindowsNT(); //判断系统
BOOL IsWindows2k();
BOOL GetUserSid(PSID* ppSid);
void GetSidString(PSID pSid, LPTSTR szBuffer);
BOOL GetOldSD(HKEY hKey, LPCTSTR pszSubKey, BYTE** pSD);
BOOL CreateNewSD(PSID pSid, SECURITY_DESCRIPTOR* pSD, PACL* ppDacl);
BOOL RegSetPrivilege(HKEY hKey, LPCTSTR pszSubKey, SECURITY_DESCRIPTOR* pSD);
BOOL WipeFile(LPCTSTR szDir, LPCTSTR szFile);
```

现在我们可以分别实现每项清除功能。

清除IE临时文件

```
TCHAR szPath[MAX_PATH];
DeleteUrlCache(File);
if (SHGetSpecialFolderPath(NULL, szPath, CSIDL_INTERNET_CACHE, FALSE)
{ //得到临时目录，并清空它。
EmptyDirectory(szPath);
}
```

清除IE临时文件的另外一个途径是直接调用Wininet函数，请看下面用于清除IE临时文件的函数

```
BOOL DelTempFiles()
{
    BOOL bResult = FALSE;
    BOOL bDone = FALSE;
    LPINTERNET_CACHE_ENTRY_INFO lpCacheEntry = NULL;

    DWORD dwTrySize, dwEntrySize = 4096; // start buffer size
    HANDLE hCacheDir = NULL;
    DWORD dwError = ERROR_INSUFFICIENT_BUFFER;

    do
    {
```

```

switch (dwError)
{
    // need a bigger buffer
    case ERROR_INSUFFICIENT_BUFFER:
        delete [] lpCacheEntry;
        lpCacheEntry = (LPINTERNET_CACHE_ENTRY_INFO) new char
        lpCacheEntry->dwStructSize = dwEntrySize;
        dwTrySize = dwEntrySize;
        BOOL bSuccess;
        if (hCacheDir == NULL)

            bSuccess = (hCacheDir
                = FindFirstUrlCacheEntry(NULL, lpCacheEntry,
                    &dwTrySize)) != NULL;
        else
            bSuccess = FindNextUrlCacheEntry(hCacheDir, lpCac

        if (bSuccess)
            dwError = ERROR_SUCCESS;
        else
        {
            dwError = GetLastError();
            dwEntrySize = dwTrySize; // use new size returned
        }
        break;

    // we are done
    case ERROR_NO_MORE_ITEMS:
        bDone = TRUE;
        bResult = TRUE;
        break;

    // we have got an entry
    case ERROR_SUCCESS:

        // don't delete cookie entry
        if (!(lpCacheEntry->CacheEntryType & COOKIE_CACHE_ENT

            DeleteUrlCacheEntry(lpCacheEntry->lpszSourceUrlName)

        // get ready for next entry
        dwTrySize = dwEntrySize;
        if (FindNextUrlCacheEntry(hCacheDir, lpCacheEntry, &c
            dwError = ERROR_SUCCESS;

        else
        {
            dwError = GetLastError();
            dwEntrySize = dwTrySize; // use new size returned
        }
        break;

    // unknown error
    default:
        bDone = TRUE;
        break;
}

if (bDone)
{
    delete [] lpCacheEntry;
    if (hCacheDir)
        FindCloseUrlCache(hCacheDir);
}

```

```

    } while (!bDone);
    return bResult;
}

```

Cookie的清除

```

TCHAR szPath[MAX_PATH];
DeleteUrlCache(Cookie);
if (SHGetSpecialFolderPath(NULL, szPath, CSIDL_COOKIES, FALSE))
{//得到目录，并清空
    EmptyDirectory(szPath);
}

```

浏览器地址栏历史地址的清除

```

SHDeleteKey(HKEY_CURRENT_USER,
    _T("Software\\Microsoft\\Internet Explorer\\TypedURLs"));

```

清除表单自动完成历史记录

```

CString sKey;
DWORD dwRet;

if (IsWindows2k() || IsWindowsNT())//先判断系统
{
    CString sBaseKey;
    SECURITY_DESCRIPTOR NewSD;
    BYTE* pOldSD;
    PACL pDacl = NULL;
    PSID pSid = NULL;
    TCHAR szSid[256];
    if (GetUserSid(&pSid))
    {
        //get the hidden key name
        GetSidString(pSid, szSid);

        sKey = _T("Software\\Microsoft\\Protected Storage System Provider\\")
        sKey += szSid;

        //get old SD
        sBaseKey = sKey;
        GetOldSD(HKEY_CURRENT_USER, sBaseKey, &pOldSD);

        //set new SD and then clear
        if (CreateNewSD(pSid, &NewSD, &pDacl))
        {
            RegSetPrivilege(HKEY_CURRENT_USER, sKey, &NewSD, FALSE);

            sKey += _T("\\Data");
            RegSetPrivilege(HKEY_CURRENT_USER, sKey, &NewSD, FALSE);

            sKey += _T("\\e161255a-37c3-11d2-bcaa-00c04fd929db");
            RegSetPrivilege(HKEY_CURRENT_USER, sKey, &NewSD, TRUE);

            dwRet = SHDeleteKey(HKEY_CURRENT_USER, sKey);
        }

        if (pDacl != NULL)
            HeapFree(GetProcessHeap(), 0, pDacl);
    }
}

```

```
//restore old SD
if (pOldSD)
{
    RegSetPrivilege(HKEY_CURRENT_USER, sBaseKey,
        (SECURITY_DESCRIPTOR*)pOldSD, FALSE);
    delete pOldSD;
}
}
if (pSid)
    HeapFree(GetProcessHeap(), 0, pSid);
}

//win9x
DWORD dwSize = MAX_PATH;
TCHAR szUserName[MAX_PATH];
GetUserName(szUserName, &dwSize);

sKey = _T("Software\\Microsoft\\Protected Storage System Provider\\")
sKey += szUserName;
sKey += _T("\\Data\\e161255a-37c3-11d2-bcaa-00c04fd929db");
dwRet = SHDeleteKey(HKEY_LOCAL_MACHINE, sKey);
```

清除自动密码历史记录

```
SHDeleteKey(HKEY_CURRENT_USER,
    _T("Software\\Microsoft\\Internet Explorer\\IntelliForms"));
```

清除收藏夹中的内容

```
TCHAR szPath[MAX_PATH];
if (SHGetSpecialFolderPath(NULL, szPath, CSIDL_FAVORITES, FALSE))
{ //得到目录，并清空
    EmptyDirectory(szPath);
}
```

清除RAS自动拨号历史记录

```
SHDeleteKey(HKEY_CURRENT_USER,
    _T("Software\\Microsoft\\RAS Autodial\\Addresses"));
```

清除系统临时文件夹

```
TCHAR szPath[MAX_PATH];
if (GetTempPath(MAX_PATH, szPath))//得到系统临时目录
{
    EmptyDirectory(szPath, TRUE);
}
```

清空回收站

```
SHEmptyRecycleBin(NULL, NULL,
    SHERB_NOCONFIRMATION | SHERB_NOPROGRESSUI | SHERB_NOSOUND);
```

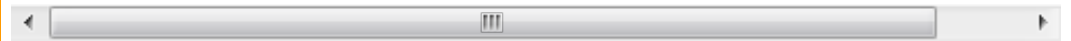
清除"运行"中的自动匹配历史记录

```
SHDeleteKey(HKEY_CURRENT_USER,
            _T("Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\F
```



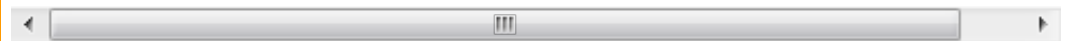
清"文档"中的历史记录

```
TCHAR szPath[MAX_PATH];
if (SHGetSpecialFolderPath(NULL, szPath, CSIDL_RECENT, FALSE))
{
    EmptyDirectory(szPath);
}
SHDeleteKey(HKEY_CURRENT_USER,
            _T("Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\RecentDoc
```



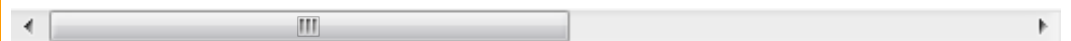
清除上次登陆用户记录

```
SHDeleteValue(HKEY_CURRENT_USER,
              _T("Software\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon
              _T("DefaultUserName"));
SHDeleteValue(HKEY_CURRENT_USER,
              _T("Software\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon
              _T("AltDefaultUserName"));
SHDeleteValue(HKEY_LOCAL_MACHINE,
              _T("Software\\Microsoft\\Windows\\CurrentVersion\\Winlogon"),
              _T("DefaultUserName"));
```



清除"查找文件"自动匹配历史记录

```
SHDeleteKey(HKEY_CURRENT_USER,
            _T("Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\D
            MRU"));
SHDeleteKey(HKEY_CURRENT_USER,
            _T("Software\\Microsoft\\Internet Explorer\\Explorer Bars\\{C
SHDeleteKey(HKEY_CURRENT_USER,
            _T("Software\\Microsoft\\Internet Explorer\\Explorer Bars\\{C
```



清除"查找计算机"自动匹配历史记录

```
SHDeleteKey(HKEY_CURRENT_USER,
            _T("Software\\Microsoft\\Windows\\CurrentVersion\\Explorer\\F
SHDeleteKey(HKEY_CURRENT_USER,
            _T("Software\\Microsoft\\Internet Explorer\\Explorer Bars\\{C
```



清除网络联接历史记录

```
TCHAR szPath[MAX_PATH];
if (SHGetSpecialFolderPath(NULL, szPath, CSIDL_NETHOOD, FALSE))
{ //得到目录，并清空
    EmptyDirectory(szPath);
}
```

清远程登录历史记录

```
CString sKey;
for (int i=1; i<=8; i++)
{
    sKey.Format(_T("Machine%d"), i);
    SHDeleteValue(HKEY_CURRENT_USER,
        _T("Software\\Microsoft\\Telnet"), sKey);

    sKey.Format(_T("Service%d"), i);
    SHDeleteValue(HKEY_CURRENT_USER,
        _T("Software\\Microsoft\\Telnet"), sKey);

    sKey.Format(_T("TermType%d"), i);
    SHDeleteValue(HKEY_CURRENT_USER,
        _T("Software\\Microsoft\\Telnet"), sKey);
}
SHDeleteValue(HKEY_CURRENT_USER,
    _T("Software\\Microsoft\\Telnet"), _T("LastMachine"));
SHDeleteValue(HKEY_CURRENT_USER,
    _T("Software\\Microsoft\\Telnet"), _T("LastService"));
SHDeleteValue(HKEY_CURRENT_USER,
    _T("Software\\Microsoft\\Telnet"), _T("LastTermType"));
```

清浏览网址历史记录

下面这个函数可以用于清除网址历史记录

```
#include "SHLGUID.H"
HRESULT ClearHistory()
{
    HRESULT hr;
    CoInitialize(NULL);
    {
        IUrlHistoryStg2* pUrlHistoryStg2 = NULL;
        hr= CoCreateInstance(CLSID_CUrlHistory,NULL,1, IID_IUrlHistorySt
        if (SUCCEEDED(hr))
        {
            hr = pUrlHistoryStg2->ClearHistory();
            pUrlHistoryStg2->Release();
        }
    }
    CoUninitialize();
    return hr;
}
```

如果上面代码不能清，则有下面的，不完美，但能工作。

```
TCHAR szPath[MAX_PATH];
GetWindowsDirectory(szPath, MAX_PATH);
_tcscat(szPath, _T("\\History"));
EmptyDirectory(szPath, FALSE, TRUE);

if (SHGetSpecialFolderPath(NULL, szPath, CSIDL_HISTORY, FALSE))
{
    EmptyDirectory(szPath, FALSE, TRUE);
}
```

好了，一个简单的清除历史程序就写完了，具体细节可以分析作者提供的源代码。

为这篇文章评分！ 很烂



很棒

评分

评论

[查看全部评论](#)

发表评论

标题

Re:

内容

发表评论

以上评论仅表达网友个人观点，不代表本网立场和观点.发表评论请遵守国家
有关法规.

[网站简介](#) | [广告服务](#) | [联系我们](#) |

Copyright © [HelloCpp](#), 2009
All Rights Reserved.
[京ICP备09051149号](#)