Click Here

The built-in Windows Remote Desktop Connection client saves the remote computer name or IP address and the username that is used to log in after each successful connection to the remote computer. This makes the next logon effortless because the RDP client offers you the possibility to select one of the connections that were used previously. All you have to do is enter a password to log in. The Server sub-key contains a list of all RDP servers and usernames used to log in to the remote terminal. In this article, we will discuss how to Remove saved RDP connections in Windows. You may want to see the following articles: How to allow saved credentials for RDP connection, how to prevent the saving of Remote Desktop Credentials in Windows, and Remote Desktop can not find the computer FQDN and this might mean that FQDN does not belong to the specified network. Information about all RDP (terminal) sessions is stored individually in the registry hive of each user. Only Domain Admins will be able to view the RDP connection history of another user. Here you will find a list of Ten (10) IP Addresses or FQDN of Remote Servers you have connected to in the past. To delete these entries, select the server sub-key and delete them. You can only delete each sub-key one after the order. HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers For other topics on RDP, see the following hyperlinks: How to disconnect a Remote Desktop User. How to clear (remove) histories from the Remote Desktop Connection. Note: In addition to the specified registry keys, you need to delete the default rdp connection file and this contains information about the latest RDP session) stored in Default.rdp (this file is a hidden file located in the Documents directory, Alternatively, Windows also saves the recent Remote Desktop connections in Jump Lists. If you type mstsc in the Windows 10 search box, the previously used RDP connections will appear in the list. You can completely disable Windows 10 recent file and location in the Jump list with the registry dword parameter Start_TrackDocs in the reg key. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced (set it to 0), or you can clear the Recent Items lists by deleting files in the directory %AppData%\Microsoft\Windows\Recent\AutomaticDestinations. Click on delete! I hope you found this blog post helpful on how to Remove saved RDP connections. Please let me know in the comment session if you have any questions. When you use the Remote Desktop Connection Tool in Windows to make a connection to another computer, the name of the computer to which you connected is added to the Remote Desktop Connection Computer box. It is done to make things easy for you. The next time you want to connect, you can easily select the computer. Over time, the list of such entries may increase, and you may want to delete them. The Windows Remote Desktop Connection Tool does not offer any way to delete or remove this history list.Remote Desktop Connection Clear HistoryIf you wish to delete them, you will either have to use the Windows Registry, a free tool, or a few more tools that work.Use RegistryRun Script to delete the historyDelete from Windows Credential ManagerDelete the Default.RDP fileUse Free tool.You will need admin privilege to get this done.Remove IP addresses from Remote Desktop Connection History in Windows 11/101] Use RegistryIf you want to manually delete the entries from the Remote Desktop Connection Computer box in the Windows Remote Desktop Connection client, type regedit and hit Enter to open the Registry Editor.Navigate to the following key:HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\DefaultThe entries will appear as MRUnumber, and are visible in the right pane. Right-click the entry and select Delete.2] Run Script to delete the historyYou can also save the below commands in a script file and then execute it. It will remove all traces of any connection made from history.@echo off reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default" /va /f reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" /f reg add "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" cd %userprofile%\documents\ attrib Default.rdp -s -h del Default.rdpOnce done, make sure to uncheck the box which asks you to remember the password and details.3] Delete from Windows Credential ManagerOpen Windows Search using Win + S, and type Credential Manager. Once it appears in the list, open it.Switch top Windows CredentialsNext, you will need to search with the name or IP address you connect on RDPThen click on the Remove link to remove it from the list.4] Delete the Default.RDP fileIt is available in the Documents folder and is a hidden file. First, disable hidden view from the File Explorer menu, and then delete the file. This will remove any latest info on the latest RDP session.5] Use Free toolA freeware called Remote Desktop History AutoCleaner also lets you do the same. It is available here.Remote Desktop History AutoCleaner monitors every Remote Desktop connection and cleans connection history. It works in the background and is lightweight.How to remove entries in the Mac Remote Desktop Connection clientTo remove entries from the Remote Desktop Connection Computer box in the Mac Remote Desktop Connection client, delete this file:Press Command key and space bar to bring the finderGo to Users:Username:Library:Preferences:Microsoft:RDC Client:Recent ServersDelete the file.Can someone access my Remote Desktop Connection without my knowledge?Yes, they can but only when they access your account. One user cannot access RDP details from another account and not even an admin account.How do I disable Remote Access to my PC?Open Windows Search (Win + S), and type Remote Desktop Settings. Then toggle off the Remote desktop option.If you want to disable Remote Desktop for another user, click on Remote Desktop users, and then remove the username from the account.How do I recover my remote desktop password?The RDP file is a configuration file that also stores your password. You can use the NirSoft Remote Desktop Pass View to view the password.Read next: How to Change RDP Listening Port in Windows. The built-in Windows Remote Desktop Connection client (mstsc.exe) saves the remote computer name (or IP address) and the username that is used to log in after each successful connection to the remote RDP host. The next time the RDP client starts, the user can select one of the saved connections, and the previous username is automatically inserted. If you are using a public computer or a computer you don't trust, you may want to clear your RDP connection history for security reasons.This article describes where Windows stores Remote Desktop connection history and credentials, and how to clear RDP history and logs.Delete the History of Remote Desktop Connections on WindowsWindows stores the history of Remote Desktop client connections in several different places, and to completely clear the RDP history you will need to delete the data from all of these places.How to Delete RDP Connection History from RegistryOpen the Registry Editor (regedit.exe) and go to the reg key HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client;The Default subkey contains history entries for the last 10 RDP connections. The names of the parameters are MRU0-MRU9 (MRU – Most Recently Used). Their values contain IP addresses/RDP host names. Select all the registry parameters in the key and delete them;Then expand the Servers subkey. A list of all previously used RDP hosts and user accounts is stored here. The username is contained in the UsernameHint parameter. This username is automatically inserted into the mstsc.exe client window when you connect to this RDP host. The CertHash parameter contains the thumbprint of the server's RDP certificate (see: "Configuring trusted TLS/SSL certificates for RDP"); Clear all entries in the Servers registry key. The easiest way to do this is to simply delete the entire Servers reg key, and then re-create it manually.Delete the Default.RDP FileThen delete the hidden Default.rdp file from the %userprofile%\Documents folder in the user profile. This file stores information about the last RDP connection.How to View and Clear RDP Event Viewer Logs in WindowsThe RDP client also logs each outbound connection to the Event Viewer log (Applications and Services Logs -> Microsoft -> Windows -> TerminalServices-ClientActiveXCore -> Microsoft-Windows-TerminalServices-RDPClient/Operational).To list the log for outbound RDP client connections, use the Get-WinEvent cmdlet to select and filter the events in the Event Viewer:$properties = @(@{n='TimeStamp';e={$_.TimeCreated}} @{n='LocalUser';e={$_.UserID}} @{n='Target RDP host';e={$_.Properties[1].Value}}) Get-WinEvent -FilterHashTable @{LogName='Microsoft-Windows-TerminalServices-RDPClient/Operational';ID='1102'} | Select-Object $propertiesYou can clear this event log from the Event Viewer console or by using the command:WevtUtil cl Microsoft-Windows-TerminalServices-RDPClient/OperationalDelete Recent RDP History Entries from the Start Menu and TaskbarWindows also stores recent remote desktop connections in Jump Lists. If you type mstsc in the Windows search box or right-click on the client in the taskbar, you will see the history of previous RDP connections in the Recent list.To clear the RDP history in the Start Menu and Jump Lists, clear the Recent Items list by deleting the files in the %AppData%\Microsoft\Windows\Recent\AutomaticDestinationsdirectory.Script to Clear RDP Connection HistoryTo quickly clear the history of RDP connections in Windows, you can use a batch script. This script will automatically perform the manual actions described above to clear the connection history and the RDP logs.@echo off reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default" /va /f reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" /f reg add "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" attrib -s -h %userprofile%\documents\Default.rdp del %userprofile%\documents\Default.rdp del /f /s /q /a %AppData%\Microsoft\Windows\Recent\AutomaticDestinationsA similar PowerShell script to clear all entries in the RDP connection history:Get-ChildItem "HKCU:\Software\Microsoft\Terminal Server Client" -Recurse | Remove-ItemProperty -Name UsernameHint -Ea 0 Remove-Item -Path 'HKCU:\Software\Microsoft\Terminal Server Client\servers' -Recurse 2>&1 | Out-Null Remove-ItemProperty -Path 'HKCU:\Software\Microsoft\Terminal Server Client\Default' 'MR*' 2>&1 | Out-Null $docs = [environment]::getfolderpath("mydocuments") + '\Default.rdp' remove-item $docs -Force 2>&1 | Out-NullHow to Clear Remote Desktop Bitmap Cache?By default, the mstsc.exe client caches rarely modified areas of the remote desktop as bitmaps (persistent bitmap caching). The RDP client cache can significantly reduce network traffic.The RDP client cache is stored in *.bmc and *.bin files in the %LOCALAPPDATA%\Microsoft\Terminal Server Client\Cachefolder.The RDP cache contains raw bitmaps of the RDP screen in the form of 64×64 pixel tiles. Attackers can use simple PowerShell or Python scripts (easily searched using the RDP Cached Bitmap Extractor query) to capture large areas of the remote desktop from cached PNGs.For security reasons, it is recommended that you clear the RDP cache folder and prevent the RDP client from saving the screen image to the cache. Disable the Persistent bitmap caching option on the Advanced tab of the Remote Desktop Connection client.How to Prevent Windows from Saving RDP Connection HistoryIf you do not want Windows to store the history of RDP connections, you must deny writing to the HKCU\Software\Microsoft\Terminal Server Client registry key. The first step is to disable the inheritance of permissions on the specified reg key (Permissions -> Advanced -> Disable inheritance). Then change the ACL of the registry key by selecting the Deny for users (but you should be aware that this is an unsupported configuration).As a result, mstsc.exe is simply not able to write any RDP connection information to the registry.Remove Saved RDP Credentials on WindowsThe RDP client allows you to store the user's password in the built-in Windows Credential Manager and automatically connect to a remote host without entering a password.You can delete saved RDP passwords from the Credential Manager window (run the command rundll32.exe keymgr.dll,KRShowKeyMgr). Remove all entries prefixed TERMSRV\.Or you can clear the stored credentials for RDP with the command:For /F "tokens=1,2 delims= " %G in ('cmdkey /list ^| findstr "target=TERMSRV"') do cmdkey /delete %H Tree-like structure for data "Hierarchical model" redirects here. For the statistics usage, see Multilevel model and Bayesian network. A hierarchical database model is a data model in which the data is organized into a tree-like structure. The data are stored as records which is a collection of one or more fields. Each field contains a single value, and the collection of fields in a record defines its type. One type of field is the link, which connects a given record to associated records. Using links, records link to other records, and to other records, forming a tree. An example is a "customer" record that has links to that customer's "orders", which in turn link to "line_items". The hierarchical database model mandates that each child record has only one parent, whereas each parent record can have zero or more child records. The network model extends the hierarchical by allowing multiple parents and children. In order to retrieve data from these databases, the whole tree needs to be traversed starting from the root node. Both models were well suited to data that was normally stored on tape drives, which had to move the tape from end to end in order to retrieve data. When the relational database model emerged, one criticism of hierarchical database models was their close dependence on application-specific implementation. This limitation, along with the relational model's ease of use, contributed to the popularity of relational databases, despite their initially lower performance in comparison with the existing network and hierarchical models.[1] The hierarchical structure was developed by IBM in the 1960s and used in early mainframe DBMS. Records' relationships form a treelike model. This structure is simple but inflexible because the relationship is confined to a one-to-many relationship. The IBM Information Management System (IMS) and RDM Mobile are examples of a hierarchical database system with multiple hierarchies over the same data. The hierarchical data model lost traction as Codd's relational model became the de facto standard used by virtually all mainstream database management systems. A relational-database implementation of a hierarchical model was first discussed in published form in 1992[2] (see also nested set model). Hierarchical data organization schemes resurfaced with the advent of XML in the late 1990s[3] (see also XML database). The hierarchical structure is used primarily today for storing geographic information and file systems.[citation needed] Currently, hierarchical databases are still widely used especially in applications that require very high performance and availability such as banking, health care, and telecommunications. One of the most widely used commercial hierarchical databases is IMS.[4] Another example of the use of hierarchical databases is Windows Registry in the Microsoft Windows operating systems.[5] An organization could store employee information in a table that contains attributes/columns such as employee number, first name, last name, and department number. The organization provides each employee with computer hardware as needed, but computer equipment may only be used by the employee to which it is assigned. The organization could store the computer hardware information in a separate table that includes each part's serial number, type, and the employee that uses it. The tables might look like this: employee table EmpNo First Name Last Name Dept. Num 100 Almukhtar Khan 10-L 101 Gaurav Soni 10-L 102 Siddhartha Soni 20-B 103 Siddhant Soni 20-B computer table Serial Num Type User EmpNo 3009734-4 Computer 100 3-23-283742 Monitor 100 2-22-723423 Monitor 100 232342 Printer 100 In this model, the employee data table represents the "parent" part of the hierarchy, while the computer table represents the "child" part of the hierarchy. In contrast to tree structures usually found in computer software algorithms, in this model the children point to the parents. As shown, each employee may possess several pieces of computer equipment, but each individual piece of computer equipment may have only one employee owner. Consider the following structure: EmpNo Designation ReportsTo 10 Director 20 Senior Manager 10 30 Typist 20 40 Programmer 20 In this, the "child" is the same type as the "parent". The hierarchy stating EmpNo 10 is boss of 20, and 30 and 40 each report to 20 is represented by the "ReportsTo" column. In Relational database terms, the ReportsTo column is a foreign key referencing the EmpNo column. If the "child" data type were different, it would be in a different table, but there would still be a foreign key referencing the EmpNo column of the employees table. This simple model is commonly known as the adjacency list model and was introduced by Dr. Edgar F. Codd after initial criticisms surfaced that the relational model could not model hierarchical data.[citation needed] However, the model is only a special case of a general adjacency list for a graph. Tree structure Hierarchical query Hierarchical clustering ^ Silberschatz, Abraham; Korth, Henry F.; Sudarshan, S. Database System Concepts. 4th ed., McGraw-Hill, 2004, p. 11, 21. ^ Michael J. Kamfonas/Recursive Hierarchies: The Relational Taboo! Archived 2008-11-08 at the Wayback Machine--The Relation Journal, October/November 1992 ^ "Web Application Development". IBM. ^ IBM Information Management System ^ "Structure of the Registry - Win32 apps". Wikimedia Commons has media related to Hierarchical models. Troels' links to Hierarchical data in RDBMSs Managing Hierarchical Data in MySQL (This page is from archive.org as the page has been removed from MySQL.com) Hierarchical data in MySQL: parents and children in one query Create Hierarchy Chart from Hierarchical Database Retrieved from " This tutorial will show you how to remove entries from the Computer box in the Remote Desktop Connection client tool for your account in Windows 10 and Windows 11. You can use Remote Desktop to connect to and control your PC from a remote device by using a Microsoft Remote Desktop client (available for Windows, iOS, macOS and Android). When you allow remote connections to your PC, you can use another device to connect to your PC and have access to all of your apps, files, and network resources as if you were sitting at your desk. After you use the Remote Desktop Connection tool to connect to another computer, the name of the computer is added to the Remote Desktop Connection Computer box. It's easy for you to quickly select the same computer at a later time. However, the tool doesn't provide a way to clear the list of computers or remove one or more entries from the Computer box. Reference: Option One: Remove All Computer Entries from Remote Desktop Connection client Option Two: Remove Specific Computer Entries from Remote Desktop Connection client EXAMPLE: Computer entries in Remote Desktop Connection client 1 Click/tap on the Download button below to download the file below. Clear_all-computer_entries_in_Remote_Desktop_Connection_cliet_for_current_user.zip (Content of BAT file for reference) :: Clear entries from RDC Cumputer box REG DELETE "HKCU\Software\Microsoft\Terminal Server Client" /F :: Delete last connection history to not show in RDC Computer box DEL /F /S /Q /A "%UserProfile%\Documents\Default.rdp" :: Clear RDC jump list DEL /F /S /Q /A "%AppData%\Microsoft\Windows\Recent\AutomaticDestinations\1bc392b8e104a00e.automaticDestinations-ms" 2 SSave the ZIP file to your desktop. 3 Unblock the ZIP file. 4 Extract the BAT file from the ZIP file. 5 Run the BAT file. 6 You will now notice a command prompt quickly open and close running the commands in the BAT file to apply. 7 You can now delete the downloaded .bat file if you like. 8 The BAT file does not delete any saved credentials of remote computers. You will need to manually delete any saved credentials you want for remote desktop connections. 1 Open Remote Desktop Connection (mstsc.exe). 2 Select the remote computer connection (ex: "Brink-Spectre") you want to remove in the Computer drop menu, and click/tap on the delete link (if available) to delete its saved credentials. (see screenshot below) 3 Click/tap on Yes to confirm deleting the saved credentials of the selected computer (ex: "Brink-Spectre"). (see screenshot below) 4 Right click or press and hold on the Remote Desktop Connection icon on the taskbar, and unpin (if pinned) the computer (ex: "Brink-Spectre") you want to remove from the jump list. (see screenshot below) 5 Right click or press and hold on the Remote Desktop Connection icon on the taskbar, right click or press and hold on the computer (ex: "Brink-Spectre") you want to remove from the jump list, and click/tap on Remove from this list. (see screenshot below) 6 Close the Remote Desktop Connection client. 7 Open Registry Editor (regedit.exe). 8 Navigate to the key below in the left pane of Registry Editor. (see screenshot below step 9) HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default 9 In the right pane of the Default key, right click on the MRU# string value (REG_SZ) for the computer (ex: "Brink-Spectre") you want to remove from RDC, and click/tap on Delete. (see screenshot below) 10 Click/tap on Yes to confirm deleting the string value (REG_SZ). (see screenshot below) 11 Navigate to the key below in the left pane of Registry Editor. (see screenshot below step 9) HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers 12 In the left pane under the Servers key, right click on the subkey with the same name as the computer (ex: "Brink-Spectre") you want to remove from RDC, and click/tap on Delete. (see screenshot below) 13 Click/tap on Yes to confirm deleting the key. (see screenshot below) 14 You can now close Registry Editor. 15 Open a command prompt. 16 Copy and paste the command below into the command prompt, and press Enter. (see screenshot below) This command will delete the last connection history so it will not continue to show in the RDC Computer box. DEL /F /S /Q /A "%UserProfile%\Documents\Default.rdp" 17 You can now close the command prompt. That's it, Shawn Brink Posts 24,647 Reaction score 20,470 Points 365 Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. The built-in Windows Remote Desktop Connection client (mstsc.exe) saves the remote computer name (or IP address) and the username that is used to log in after each successful connection to the remote RDP host. The next time the RDP client starts, the user can select one of the saved connections, and the previous username is automatically inserted. If you are using a public computer or a computer you don't trust, you may want to clear your RDP connection history for security reasons.This article describes where Windows stores Remote Desktop connection history and credentials, and how to clear RDP history and logs.Delete the History of Remote Desktop Connections on WindowsWindows stores the history of Remote Desktop client connections in several different places, and to completely clear the RDP history you will need to delete the data from all of these places.How to Delete RDP Connection History from RegistryOpen the Registry Editor (regedit.exe) and go to the reg key HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client;The Default subkey contains history entries for the last 10 RDP connections. The names of the parameters are MRU0-MRU9 (MRU – Most Recently Used). Their values contain IP addresses/RDP host names. Select all the registry parameters in the key and delete them;Then expand the Servers subkey. A list of all previously used RDP hosts and user accounts is stored here. The username is contained in the UsernameHint parameter. This username is automatically inserted into the mstsc.exe client window when you connect to this RDP host. The CertHash parameter contains the thumbprint of the server's RDP certificate (see: "Configuring trusted TLS/SSL certificates for RDP"); Clear all entries in the Servers registry key. The easiest way to do this is to simply delete the entire Servers reg key, and then re-create it manually.Delete the Default.RDP FileThen delete the hidden Default.rdp file from the %userprofile%\Documents folder in the user profile. This file stores information about the last RDP connection.How to View and Clear RDP Event Viewer Logs in WindowsThe RDP client also logs each outbound connection to the Event Viewer log (Applications and Services Logs -> Microsoft -> Windows -> TerminalServices-ClientActiveXCore -> Microsoft-Windows-TerminalServices-RDPClient/Operational).To list the log for outbound RDP client connections, use the Get-WinEvent cmdlet to select and filter the events in the Event Viewer:$properties = @(@{n='TimeStamp';e={$_.TimeCreated}} @{n='LocalUser';e={$_.UserID}} @{n='Target RDP host';e={$_.Properties[1].Value}}) Get-WinEvent -FilterHashTable @{LogName='Microsoft-Windows-TerminalServices-RDPClient/Operational';ID='1102'} | Select-Object $propertiesYou can clear this event log from the Event Viewer console or by using the command:WevtUtil cl Microsoft-Windows-TerminalServices-RDPClient/OperationalDelete Recent RDP History Entries from the Start Menu and TaskbarWindows also stores recent remote desktop connections in Jump Lists. If you type mstsc in the Windows search box or right-click on the client in the taskbar, you will see the history of previous RDP connections in the Recent list.To clear the RDP history in the Start Menu and Jump Lists, clear the Recent Items list by deleting the files in the %AppData%\Microsoft\Windows\Recent\AutomaticDestinationsdirectory.Script to Clear RDP Connection HistoryTo quickly clear the history of RDP connections in Windows, you can use a batch script. This script will automatically perform the manual actions described above to clear the connection history and the RDP logs.@echo off reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default" /va /f reg delete "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" /f reg add "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Servers" attrib -s -h %userprofile%\documents\Default.rdp del %userprofile%\documents\Default.rdp del /f /s /q /a %AppData%\Microsoft\Windows\Recent\AutomaticDestinationsA similar PowerShell script to clear all entries in the RDP connection history:Get-ChildItem "HKCU:\Software\Microsoft\Terminal Server Client" -Recurse | Remove-ItemProperty -Name UsernameHint -Ea 0 Remove-Item -Path 'HKCU:\Software\Microsoft\Terminal Server Client\servers' -Recurse 2>&1 | Out-Null Remove-ItemProperty -Path 'HKCU:\Software\Microsoft\Terminal Server Client\Default' 'MR*' 2>&1 | Out-Null $docs = [environment]::getfolderpath("mydocuments") + '\Default.rdp' remove-item $docs -Force 2>&1 | Out-NullHow to Clear Remote Desktop Bitmap Cache?By default, the mstsc.exe client caches rarely modified areas of the remote desktop as bitmaps (persistent bitmap caching). The RDP client cache can significantly reduce network traffic.The RDP client cache is stored in *.bmc and *.bin files in the %LOCALAPPDATA%\Microsoft\Terminal Server Client\Cachefolder.The RDP cache contains raw bitmaps of the RDP screen in the form of 64×64 pixel tiles. Attackers can use simple PowerShell or Python scripts (easily searched using the RDP Cached Bitmap Extractor query) to capture large areas of the remote desktop from cached PNGs.For security reasons, it is recommended that you clear the RDP cache folder and prevent the RDP client from saving the screen image to the cache. Disable the Persistent bitmap caching option on the Advanced tab of the Remote Desktop Connection client.How to Prevent Windows from Saving RDP Connection HistoryIf you do not want Windows to store the history of RDP connections, you must deny writing to the HKCU\Software\Microsoft\Terminal Server Client registry key. The first step is to disable the inheritance of permissions on the specified reg key (Permissions -> Advanced -> Disable inheritance). Then change the ACL of the registry key by selecting the Deny for users (but you should be aware that this is an unsupported configuration).As a result, mstsc.exe is simply not able to write any RDP connection information to the registry.Remove Saved RDP Credentials on WindowsThe RDP client allows you to store the user's password in the built-in Windows Credential Manager and automatically connect to a remote host without entering a password.You can delete saved RDP passwords from the Credential Manager window (run the command rundll32.exe keymgr.dll,KRShowKeyMgr). Remove all entries prefixed TERMSRV\.Or you can clear the stored credentials for RDP with the command:For /F "tokens=1,2 delims= " %G in ('cmdkey /list ^| findstr "target=TERMSRV"') do cmdkey /delete %H MiniTool NewsUpdates and announcements about MiniTool. MiniTool Power Data Recovery 12.5: Smarter File Scanning and Safer Recovery MiniTool MovieMaker 8.2: Copy Individual Elements on the Timeline MiniTool ShadowMaker 4.7: Unified the interface of the bootable version with the Windows version MiniTool Video Converter 4.0: New UI design with optimized features and processes MiniTool Partition Wizard 12.9: Optimized the Data Recovery function Database for Microsoft Windows Windows RegistryRegistry Editor, the user interface for the registry, in Windows 11Developer(s)MicrosoftInitial releaseApril 6, 1992; 33 years ago (1992-04-06) with Windows 3.1Operating systemMicrosoft WindowsPlatformIA-32, x86-64 and ARM (and historically DEC Alpha, Itanium, MIPS, and PowerPC)Included withMicrosoft WindowsTypeHierarchical databaseLicenseProprietaryWebsitelearn.microsoft.com/en-us/windows/win32/sysinfo/registry The Windows Registry is a hierarchical database that stores low-level settings for the Microsoft Windows operating system and for applications that opt to use the registry. The kernel, device drivers, services, Security Accounts Manager, and user interfaces can all use the registry. The registry also allows access to counters for profiling system performance. In other words, the registry or Windows Registry contains information, settings, options, and other values for programs and hardware installed on all versions of Microsoft Windows operating systems. For example, when a program is installed, a new subkey containing settings such as a program's location, its version, and how to start the program, are all added to the Windows Registry. When introduced with Windows 3.1, the Windows Registry primarily stored configuration information for COM-based components. Windows 95 and Windows NT extended its use to rationalize and centralize the information in the profusion of INI files, which held the configurations for individual programs, and were stored at various locations.[1][2] It is not a requirement for Windows applications to use the Windows Registry. For example, .NET Framework applications use XML files for configuration, while portable applications usually keep their configuration files with their executables. Prior to the Windows Registry, .INI files stored each program's settings as a text file or binary file, often located in a shared location that did not provide user-specific settings in a multi-user scenario. By contrast, the Windows Registry stores all application settings in one logical repository (but also in a number of discrete files) and in a standardized form. According to Microsoft, this offers several advantages over .INI files.[2][3] Since file parsing is done much more efficiently with a binary format, it may be read from or written to more quickly than a text INI file. Furthermore, strongly typed data can be stored in the registry, as opposed to the text information stored in .INI files. This is a benefit when editing keys manually using regedit.exe, the built-in Windows Registry Editor. Because user-based registry settings are loaded from a user-specific path rather than from a read-only system location, the registry allows multiple users to share the same machine, and also allows programs to work for less privileged users. Backup and restoration is also simplified as the registry can be accessed over a network connection for remote management/support, including from scripts, using the standard set of APIs, as long as the Remote Registry service is running and firewall rules permit this. Because the registry is a database, it offers improved system integrity with features such as atomic updates. If two processes attempt to update the same registry value at the same time, one process's change will precede the other's and the overall consistency of the data will be maintained. Where changes are made to .INI files, such race conditions can result in inconsistent data that does not match either attempted update. Windows Vista and later operating systems provide transactional updates to the registry by means of the Kernel Transaction Manager, extending the atomicity guarantees across multiple key or value changes with traditional commit–abort semantics. (Note however that NTFS provides such support for the file system as well, so the same guarantees could, in theory, be obtained with traditional configuration files.) The registry contains two basic elements: keys and values. Registry keys are container objects similar to folders. Registry values are non-container objects similar to files. Keys may contain values and subkeys. Keys are referenced with a syntax similar to Windows' path names, using backslashes to indicate levels of hierarchy. Keys must have a case insensitive name without backslashes. The hierarchy of registry keys can only be accessed from a known root key handle (which is anonymous but whose effective value is a constant numeric handle) that is mapped to the content of a registry key preloaded by the kernel from a stored "hive", or to the content of a subkey within another root key, or mapped to a registered service or DLL that provides access to its contained subkeys and values. E.g. HKEY_LOCAL_MACHINE\Software\Microsoft\Windows refers to the subkey "Windows" of the subkey "Microsoft" of the subkey "Software" of the HKEY_LOCAL_MACHINE root key. There are seven predefined root keys, traditionally named according to their constant handles defined in the Win32 API, or by synonymous abbreviations (depending on applications):[4] HKEY_LOCAL_MACHINE or HKLM HKEY_CURRENT_CONFIG or HKCC HKEY_CLASSES_ROOT or HKCR HKEY_CURRENT_USER or HKCU HKEY_USERS or HKU HKEY_PERFORMANCE_DATA (only in Windows NT, but invisible in the Windows Registry Editor)[5] HKEY_DYN_DATA (only in Windows 9x, and visible in the Windows Registry Editor) Like other files and services in Windows, all registry keys may be restricted by access control lists (ACLs), depending on user privileges, or on security tokens acquired by applications, or on system security policies enforced by the system (these restrictions may be predefined by the system itself, and configured by local system administrators or by domain administrators). Different users, programs, services or remote systems may only see some parts of the hierarchy or distinct hierarchies from the same root keys. Registry values are name/data pairs stored within keys. Registry values are referenced separately from registry keys. Each registry value stored in a registry key has a unique name whose letter case is not significant. The Windows API functions that query and manipulate registry values take value names separately from the key path or handle that identifies the parent key. Registry values may contain backslashes in their names, but doing so makes them difficult to distinguish from their key paths when using some legacy Windows Registry API functions (whose usage is deprecated in Win32). The terminology is somewhat misleading, as each registry key is similar to an associative array, where standard terminology would refer to the name part of each registry value as a "key". The terms are a holdout from the 16-bit registry in Windows 3, in which registry keys could not contain arbitrary name/data pairs, but rather contained only one unnamed value (which had to be a string). In this sense, the Windows 3 registry was like a single associative array, in which the keys (in the sense of both 'registry key' and 'associative array key') formed a hierarchy, and the registry values were all strings. When the 32-bit registry was created, so was the additional capability of creating multiple named values per key, and the meanings of the names were somewhat distorted.[6] For compatibility with the previous behavior, each registry key may have a "default" value, whose name is the empty string. Each value can store arbitrary data with variable length and encoding, but which is associated with a symbolic type (defined as a numeric constant) defining how to parse this data. The standard types are:[7] List of standard registry value types Type ID Symbolic type name Meaning and encoding of the data stored in the registry value 0 REG_NONE No type (the stored value, if any) 1 REG_SZ A string value, normally stored and exposed in UTF-16LE (when using the Unicode version of Win32 API functions), usually terminated by a NUL character 2 REG_EXPAND_SZ An "expandable" string value that can contain environment variables, normally stored and exposed in UTF-16LE, usually terminated by a NUL character 3 REG_BINARY Binary data (any arbitrary data) 4 REG_DWORD / REG_DWORD_LITTLE_ENDIAN A DWORD value, a 32-bit unsigned integer (numbers between 0 and 4,294,967,295 [232 – 1]) (little-endian) 5 REG_DWORD_BIG_ENDIAN A DWORD value, a 32-bit unsigned integer (numbers between 0 and 4,294,967,295 [232 – 1]) (big-endian) 6 REG_LINK A symbolic link (UNICODE) to another registry key, specifying a root key and the path to the target key 7 REG_MULTI_SZ A multi-string value, which is an ordered list of non-empty strings, normally stored and exposed in Unicode, each one terminated by a null character, the list being normally terminated by a second null character.[8] 8 REG_RESOURCE_LIST A resource list (used by the Plug-n-Play hardware enumeration and configuration) 9 REG_FULL_RESOURCE_DESCRIPTOR A resource descriptor (used by the Plug-n-Play hardware enumeration and configuration) 10 REG_RESOURCE_REQUIREMENTS_LIST A resource requirements list (used by the Plug-n-Play hardware enumeration and configuration) 11 REG_QWORD / REG_QWORD_LITTLE_ENDIAN A QWORD value, a 64-bit integer (either big- or little-endian, or unspecified) (introduced in Windows 2000)[9] The keys at the root level of the hierarchical database are generally named by their Windows API definitions, which all begin with "HKEY".[2] They are frequently abbreviated to a three- or four-letter short name starting with "HK" (e.g. HKCU and HKLM). Technically, they are predefined handles (with known constant values) to specific keys that are either maintained in memory, or stored in hive files stored in the local filesystem and loaded by the system kernel at boot time and then shared (with various access rights) between all processes running on the local system, or loaded and mapped in all processes started in a user session when the user logs on the system. The HKEY_LOCAL_MACHINE (local machine-specific configuration data) and HKEY_CURRENT_USER (user-specific configuration data) nodes have a similar structure to each other; user applications typically look up their settings by first checking for them in HKEY_CURRENT_USER\Software\Vendor's name\Application's name\Version\Setting name, and if the setting is not found, look instead in the same location under the HKEY_LOCAL_MACHINE key. However, the converse may apply for administrator-enforced policy settings where HKLM may take precedence over HKCU. The Windows Logo Program has specific requirements for where different types of user data may be stored, and that the concept of least privilege be followed so that administrator-level access is not required to use an application.[a][10] Abbreviated HKLM, HKEY_LOCAL_MACHINE stores settings that are specific to the local computer.[11] The key located by HKLM is actually not stored on disk, but maintained in memory by the system kernel in order to map all the other subkeys. Applications cannot create any additional subkeys. On Windows NT, this key contains four subkeys, "SAM", "SECURITY", "SYSTEM", and "SOFTWARE", that are loaded at boot time within their respective files located in the %SystemRoot%\System32\config\ folder. A fifth subkey, "HARDWARE", is volatile and is created dynamically, and as such is not stored in a file (it exposes a view of all the currently detected Plug-and-Play devices). On Windows Vista and above, a sixth and seventh subkey, "COMPONENTS" and "BCD", are mapped in memory by the kernel on-demand and loaded from %SystemRoot%\System32\config\COMPONENTS or from boot configuration data, \boot\BCD on the system partition. The "HKLM\SAM" key usually appears as empty for most users (unless they are granted access by administrators of the local system or administrators of domains managing the local system). It is used to reference all "Security Accounts Manager" (SAM) databases for all domains into which the local system has been administratively authorized or configured (including the local domain of the running system, whose SAM database is stored in a subkey also named "SAM": other subkeys will be created as needed, one for each supplementary domain). Each SAM database contains all builtin accounts (mostly group aliases) and configured accounts (users, groups and their aliases, including guest accounts and administrator accounts) created and configured on the respective domain, for each account in that domain, it notably contains the user name which can be used to log on that domain, the internal unique user identifier in the domain, a cryptographic hash of each user's password for each enabled authentication protocol, the location of storage of their user registry hive, various status flags (for example if the account can be enumerated and be visible in the logon prompt screen), and the list of domains (including the local domain) into which the account was configured. The "HKLM\SECURITY" key usually appears empty for most users (unless they are granted access by users with administrative privileges) and is linked to the Security database of the domain into which the current user is logged on (if the user is logged on the local system domain, this key will be linked to the registry hive stored by the local machine and managed by local system administrators or by the builtin "System" account and Windows installers). The kernel will access it to read and enforce the security policy applicable to the current user and all applications or operations executed by this user. It also contains a "SAM" subkey which is dynamically linked to the SAM database of the domain onto which the current user is logged on. The "HKLM\SYSTEM" key is normally only writable by users with administrative privileges on the local system. It contains information about the Windows system setup, data for the secure random number generator (RNG), the list of currently mounted devices containing a filesystem, several numbered HKLM\SYSTEM\Control Sets containing alternative configurations for system hardware drivers and services running on the local system (including the currently used one and a backup), a "HKLM\SYSTEM\Select" subkey containing the status of these Control Sets, and a "HKLM\SYSTEM\CurrentControlSet" which is dynamically linked at boot time to the Control Set which is currently used on the local system. Each configured Control Set contains: an "Enum" subkey enumerating all known Plug-and-Play devices and associating them with installed system drivers (and storing the device-specific configurations of these drivers), a "Services" subkey listing all installed system drivers (with non device-specific configuration, and the enumeration of devices for which they are instantiated) and all programs running as services (how and when they can be automatically started), a "Control" subkey organizing the various hardware drivers and programs running as services and all other system-wide configuration, a "Hardware Profiles" subkey enumerating the various profiles that have been tuned (each one with "System" or "Software" settings used to modify the default profile, either in system drivers and services or in the applications) as well as the Hardware Profiles\Current subkey which is dynamically linked to one of these profiles. The "HKLM\SOFTWARE" subkey contains software and Windows settings (in the default hardware profile). It is mostly modified by application and system installers. It is organized by software vendor (with a subkey for each), but also contains a "Windows" subkey for some settings of the Windows user interface, a "Classes" subkey containing all registered associations from file extensions, MIME types, Object Classes IDs and interfaces IDs (for OLE, COM/DCOM and ActiveX), to the installed applications or DLLs that may be handling these types on the local machine (however these associations are configurable for each user, see below), and a "Policies" subkey (also organized by vendor) for enforcing general usage policies on applications and system services (including the central certificates store used for authenticating, authorizing or disallowing remote systems or services running outside the local network domain). The "HKLM\SOFTWARE\Wow6432Node" key is used by 32-bit applications on a 64-bit Windows OS, and is equivalent to but separate from "HKLM\SOFTWARE". The key path is transparently presented to 32-bit applications by WoW64 as HKLM\SOFTWARE[12] (in a similar way that 64-bit applications see %SystemRoot%\Syswow64 as %SystemRoot%\System32) Abbreviated HKCR, HKEY_CLASSES_ROOT contains information about registered applications, such as file associations and OLE Object Class IDs, tying them to the applications used to handle these items. On Windows 2000 and above, HKCR is a compilation of user-based HKCU\Software\Classes and machine-based HKLM\Software\Classes. If a given value exists in both of the subkeys above, the one in HKCU\Software\Classes takes precedence.[13] The design allows for either machine- or user-specific registration of COM objects. Abbreviated HKU, HKEY_USERS contains subkeys corresponding to the HKEY_CURRENT_USER keys for each user profile actively loaded on the machine, though user hives are usually only loaded for currently logged-in users. Abbreviated HKCU, HKEY_CURRENT_USER stores settings that are specific to the currently logged-in user.[14] The HKEY_CURRENT_USER key is a link to the subkey of HKEY_USERS that corresponds to the user; the same information is accessible in both locations. The specific subkey referenced is (HKU)\(SID)\... where (SID) corresponds to the Windows SID; if the "(HKCU)" key has the following suffix (HKCU)\Software\Classes\... then it corresponds to (HKU)\(SID)_CLASSES\... i.e. the suffix string "_CLASSES" is appended to the (SID). On Windows NT systems, each user's settings are stored in their own files called NTUSER.DAT and USRCLASS.DAT inside their own Documents and Settings subfolder (or their own Users sub folder in Windows Vista and above). Settings in this hive follow users with a roaming profile from machine to machine. This key provides runtime information into performance data provided by either the NT kernel itself, or running system drivers, programs and services that provide performance data. This key is not stored in any hive and not displayed in the Registry Editor, but it is visible through the registry functions in the Windows API, or in a simplified view via the Performance tab of the Task Manager (only for a few performance data on the local system) or via more advanced control panels (such as the Performances Monitor or the Performances Analyzer which allows collecting and logging these data, including from remote systems). This key is used only in Windows 95, Windows 98 and Windows ME.[15] It contains information about hardware devices, including Plug and Play and network performance statistics. The information in this hive is also not stored on the hard drive; the Plug and Play information is gathered and configured at startup and is stored in memory.[16] Even though the registry presents itself as an integrated hierarchical database, branches of the registry are actually stored in a number of disk files called hives.[17] (The word hive constitutes an in-joke.)[18] Some hives are volatile and are not stored on disk at all. An example of this is the hive of the branch starting at HKLM\HARDWARE. This hive records information about system hardware and is created each time the system boots and performs hardware detection. Individual settings for users on a system are stored in a hive (disk file) per user. During user login, the system loads the user hive under the HKEY_USERS key and sets the HKCU (HKEY_CURRENT_USER) symbolic reference to point to the current user. This allows applications to store/retrieve settings for the current user implicitly under the HKCU key. Not all hives are loaded at any one time. At boot time, only a minimal set of hives are loaded, and after that, hives are loaded as the operating system initializes and as users log in or whenever a hive is explicitly loaded by an application. The registry is physically stored in several files, which are generally obfuscated from the user-mode APIs used to manipulate the data inside the registry. Depending upon the version of Windows, there will be different files and different locations for these files, but they are all on the local machine. The location for system registry files in Windows NT is %SystemRoot%\System32\config\; the user-specific HKEY_CURRENT_USER user registry hive is stored in Ntuser.dat inside the user profile. There is one of these per user; if a user has a roaming profile, then this file will be copied to and from a server at logout and login respectively. A second user-specific registry file named UsrClass.dat contains COM registry entries and does not roam by default. Windows NT systems store the registry in a binary file format which can be exported, loaded and unloaded by the Registry Editor in these operating systems. The following system registry files are stored in %SystemRoot%\System32\config\: Sam – HKEY_LOCAL_MACHINE\SAM Security – HKEY_LOCAL_MACHINE\SECURITY Software – HKEY_LOCAL_MACHINE\SOFTWARE System – HKEY_LOCAL_MACHINE\SYSTEM Default – HKEY_USERS\.DEFAULT Userdiff – Not associated with a hive. Used only when upgrading operating systems.[19] The following file is stored in each user's profile folder: %USERPROFILE%\Ntuser.dat – HKEY_USERS\ (linked to by HKEY_CURRENT_USER) For Windows 2000, Server 2003 and Windows XP, the following additional user-specific file is used for file associations and COM information: %USERPROFILE%\Local Settings\Application Data\Microsoft\Windows\Usrclass.dat (path is localized) – HKEY_USERS_Classes (HKEY_CURRENT_USER\Software\Classes) For Windows Vista and later, the path was changed to: %USERPROFILE%\AppData\Local\Microsoft\Windows\Usrclass.dat (path is not localized) alias %LocalAppData%\Microsoft\Windows\Usrclass.dat – HKEY_USERS_Classes (HKEY_CURRENT_USER\Software\Classes) Windows 2000 keeps an alternate copy of the registry hives (.ALT) and attempts to switch to it when corruption is detected.[20] Windows XP and Windows Server 2003 do not maintain a System.alt hive because NTLDR on those versions of Windows can process the System.log file to bring up to date a System hive that has become inconsistent during a shutdown or crash. In addition, the %SystemRoot%\Repair folder contains a copy of the system's registry hives that were created after installation and the first successful startup of Windows. Each registry data file has an associated file with a ".log" extension that acts as a transaction log that is used to ensure that any interrupted updates can be completed upon next startup.[21] Internally, Registry files are split into 4 kB "bins" that contain collections of "cells".[21] The registry files are stored in the %WINDIR% directory under the names USER.DAT and SYSTEM.DAT with the addition of CLASSES.DAT in Windows ME. Also, each user profile (if profiles are enabled) has its own USER.DAT file which is located in the user's profile directory in %WINDIR%\Profiles\. The only registry file is called REG.DAT and it is stored in the %WINDIR% directory. To access the registry files, the device needs to be set in a special mode using either: WpInternals (Put the device into flash mode.) InterOp Tools (Mount the MainOS Partition with MTP.) If any of the above methods worked, the device's registry files can be found in the following location: {Phone}\EFIESP\Windows\System32\config The registry contains important configuration information for the operating system, for installed applications as well as individual settings for each user and application. A careless change to the operating system configuration in the registry could cause irreversible damage, so it is usually only installer programs which perform changes to the registry database during installation/configuration and removal. If a user wants to edit the registry manually, Microsoft recommends that a backup of the registry be performed before the change.[22] When a program is removed from control panel, it may not be completely removed and, in case of errors or glitches caused by references to missing programs, the user might have to manually check inside directories such as program files. After this, the user might need to manually remove any reference to the uninstalled program in the registry. This is usually done by using RegEdit.exe.[23] Editing the registry is sometimes necessary when working around Windows-specific issues e.g. problems when logging onto a domain can be resolved by editing the registry.[24] Windows Registry can be edited manually using programs such as RegEdit.exe, although these tools do not expose some of the registry's metadata such as the last modified date. The registry editor for the 3.1/95 series of operating systems is RegEdit.exe and for Windows NT it is RegEdt32.exe; the functionalities are merged in Windows XP. Optional and third-party tools similar to RegEdit.exe are available for many Windows CE versions. Registry Editor allows users to perform the following functions: Creating, manipulating, renaming[25] and deleting registry keys, subkeys, values and value data Importing and exporting .REG files, exporting data in the binary hive format Loading, manipulating and unloading registry hive format files (Windows NT systems only) Setting permissions based on ACLs (Windows NT systems only) Bookmarking user-selected registry keys as Favorites Finding particular strings in key names, value names and value data Remotely editing the registry on another networked computer .REG files (also known as Registration entries) are text-based human-readable files for exporting and importing portions of the registry using an INI-based syntax. On Windows 2000 and later, they contain the string Windows Registry Editor Version 5.00 at the beginning, while on Windows 9x and NT 4.0 systems, they contain the string REGEDIT4.[26] Windows 2000 and later REG files are Unicode-based, while on Windows 9x and NT 4.0 systems, they are ANSI-based. [citation needed] Windows 9x format .REG files are compatible with Windows 2000 and later.[26] The Registry Editor on Windows on these systems also supports exporting .REG files in Windows 9x/NT format.[citation needed] Data is stored in .REG files using the following syntax:[26] [\\] "Value name"=: The Default Value of a key can be edited by using @ instead of "Value Name": [\\] @=: String values do not require a (see example), but backslashes (\) need to be written as a double-backslash (\\), and quotes (") as backslash-quote (\"). For example, to add the values "Value A", "Value B", "Value C", "Value D", "Value E", "Value F", "Value G", "Value H", "Value I", "Value J", "Value K", "Value L", and "Value M" to the HKLM\SOFTWARE\Foobar key: Windows Registry Editor Version 5.00 [HKEY_LOCAL_MACHINE\SOFTWARE\Foobar] "Value A"="" "Value B"=hex: "Value C"=dword: "Value D"=hex(0): "Value E"=hex(1): "Value F"=hex(2): "Value G"=hex(3): ; equal to "Value B" "Value H"=hex(4): "Value I"=hex(5): "Value J"=hex(7): "Value K"=hex(8): "Value L"=hex(a): "Value M"=hex(b): Data from .REG files can be added/merged with the registry by double-clicking these files or using the /s switch in the command line. REG files can also be used to remove registry data. To remove a key (and all subkeys, values and data), the key name must be preceded by a minus sign (-).[26] For example, to remove the HKLM\SOFTWARE\Foobar key (and all subkeys, values and data), [-HKEY_LOCAL_MACHINE\SOFTWARE\Foobar] To remove a value (and its data), the values to be removed must have a minus sign (-) after the equal sign (=).[26] For example, to remove only the "Value A" and "Value B" values (and their data) from the HKLM\SOFTWARE\Foobar key: [HKEY_LOCAL_MACHINE\SOFTWARE\Foobar] "Value A"=- "Value B"=- To remove only the Default value of the key HKLM\SOFTWARE\Foobar (and its data): [HKEY_LOCAL_MACHINE\SOFTWARE\Foobar] @=- Lines beginning with a semicolon are considered comments: ; This is a comment. This can be placed in any part of a .reg file [HKEY_LOCAL_MACHINE\SOFTWARE\Foobar] "Value"="Example string" Main article: Group Policy Windows group policies can change registry keys for a number of machines or individual users based on policies. When a policy first takes effect for a machine or for an individual user of a machine, the registry settings specified as part of the policy are applied to the machine or user settings. Windows will also look for updated policies and apply them periodically, typically every 90 minutes.[27] Through its scope a policy defines to which machines and users the policy is to be applied. Whether a machine or user is within the scope of a policy or not is defined by a set of rules which can filter on the location of the machine or user account in organizational directory, specific users or machine accounts or security groups. More advanced rules can be set up using Windows Management Instrumentation expressions. Such rules can filter on properties such as computer vendor name, CPU architecture, installed software, or networks connected to. For instance, the administrator can create a policy with one set of registry settings for machines in the accounting department and policy with another (lock-down) set of registry settings for kiosk terminals in the visitors area. When a machine is moved from one scope to another (e.g., changing its name or moving it to another organizational unit), the correct policy is automatically applied. When a policy is changed it is automatically re-applied to all machines currently in its scope. The policy is edited through a number of administrative templates which provides a user interface for picking and changing settings. The set of administrative templates is extensible and software packages which support such remote administration can register their own templates. regDeveloper(s)MicrosoftOperating systemMicrosoft WindowsTypeCommandLicenseProprietary commercial softwareWebsitedocs.microsoft.com/en-us/windows-server/administration/windows-commands/reg reginiDeveloper(s)MicrosoftOperating systemMicrosoft WindowsTypeCommandLicenseProprietary commercial softwareWebsitedocs.microsoft.com/en-us/windows-server/administration/windows-commands/regini The registry can be manipulated in a number of ways from the command line. The Reg.exe and RegIni.exe utility tools are included in Windows XP and later versions of Windows. Alternative locations for legacy versions of Windows include the Resource Kit CDs or the original Installation CD of Windows. Also, a .REG file can be imported from the command line with the following command: RegEdit.exe /s file The /s means the file will be silent merged to the registry. If the /s parameter is omitted the user will be asked to confirm the operation. In Windows 98, Windows 95 and at least some configurations of Windows XP the /s switch also causes RegEdit.exe to ignore the setting in the registry that allows administrators to disable it. When using the /s switch RegEdit.exe does not return an appropriate return code if the operation fails, unlike Reg.exe which does. RegEdit.exe /e file exports the whole registry in V5 format to a UNICODE .REG file, while any of RegEdit.exe /e file HKEY_CLASSES_ROOT[\] RegEdit.exe /e file HKEY_CURRENT_CONFIG[\] RegEdit.exe /e file HKEY_CURRENT_USER[\] RegEdit.exe /e file HKEY_LOCAL_MACHINE[\] RegEdit.exe /e file HKEY_USERS[\] export the specified (sub)key (which has to be enclosed in quotes if it contains spaces) only. RegEdit.exe /a file exports the whole registry in V4 format to an ANSI .REG file. RegEdit.exe /a file exports the specified (sub)key (which has to be enclosed in quotes if it contains spaces) only. It is also possible to use Reg.exe. Here is a sample to display the value of the registry value Version: Reg.exe QUERY HKLM\Software\Microsoft\ResKit /v Version Other command line options include a VBScript or JScript together with CScript, WMI or WMIC.exe and Windows PowerShell. Registry permissions can be manipulated through the command line using RegIni.exe and the SubInACL.exe tool. For example, the permissions on the HKEY_LOCAL_MACHINE\SOFTWARE key can be displayed using: SubInACL.exe /keyreg HKEY_LOCAL_MACHINE\SOFTWARE /display Using PowerShell to navigate the registry Windows PowerShell comes with a registry provider which presents the registry as a location type similar to the file system. The same commands used to manipulate files and directories in the file system can be used to manipulate keys and values of the registry.[28] Also like the file system, PowerShell uses the concept of a current location which defines the context on which commands by default operate. The Get-ChildItem (also available through the aliases ls, dir or gci) retrieves the child keys of the current location. By using the Set-Location (or the alias cd) command the user can change the current location to another key of the registry.[28] Commands which rename items, remove items, create new items or set content of items or properties can be used to rename keys, remove keys or entire sub-trees or change values. Through PowerShell scripts files, an administrator can prepare scripts which, when executed, make changes to the registry. Such scripts can be distributed to administrators who can execute them on individual machines. The PowerShell Registry provider supports transactions, i.e. multiple changes to the registry can be bundled into a single atomic transaction. An atomic transaction ensures that either all of the changes are committed to the database, or if the script fails, none of the changes are committed to the database.[28][29] The registry can be edited through the APIs of the Advanced Windows 32 Base API Library (advapi32.dll).[30] List of registry API functions: RegCloseKey RegConnectRegistry RegCreateKey RegCreateKeyEx RegDeleteKey RegDeleteValue RegEnumKey RegEnumKeyEx RegEnumValue RegFlushKey RegGetKeySecurity RegLoadKey RegNotifyChangeKeyValue RegOpenKey RegOpenKeyEx RegQueryInfoKey RegQueryMultipleValues RegQueryValue RegQueryValueEx RegReplaceKey RegRestoreKey RegSaveKey RegSetKeySecurity RegSetValue RegSetValueEx RegUnLoadKey Many programming languages offer built-in runtime library functions or classes that wrap the underlying Windows APIs and thereby enable programs to store settings in the registry (e.g. Microsoft.Win32.Registry in VB.NET and C#, or TRegistry in Delphi and Free Pascal). COM-enabled applications like Visual Basic 6 can use the WSH WScript.Shell object. Another way is to use the Windows Resource Kit Tool, Reg.exe by executing it from code,[31] although this is considered poor programming practice. Similarly, scripting languages such as Perl (with Win32::TieRegistry), Python (with winreg), TCL (which comes bundled with the registry package),[32] Windows Powershell and Windows Scripting Host also enable registry editing from scripts. The offreg.dll[33] available from the Windows Driver Kit offers a set of APIs for the creation and manipulation of currently not loaded registry hives similar to those provided by advapi32.dll. It is also possible to edit the registry (hives) of an offline system from Windows PE or Linux (in the latter case using open source tools). Prior to the introduction of registration-free COM, developers were encouraged to add initialization code to in-process and out-of-process binaries to perform the registry configuration required for that object to work. For in-process binaries such as .DLL and .OCX files, the modules typically exported a function called DllInstall()[34] that could be called by installation programs or invoked manually with utilities like Regsvr32.exe;[35] out-of-process binaries typically support the commandline arguments /Regserver and /Unregserver that created or deleted the required registry settings.[36] COM applications that break because of DLL Hell issues can commonly be repaired with RegSvr32.exe or the /RegServer switch without having to re-invoke installation programs.[37] Windows exposes APIs that allows user-mode applications to register to receive a notification event if a

particular registry key is changed.[38] APIs are also available to allow kernel-mode applications to filter and modify registry calls made by other applications.[39] Windows also supports remote access to the registry of another computer via the RegConnectRegistry function[40] if the Remote Registry service is running, correctly configured and its network traffic is not firewalled.[41] Main article: Security descriptor Each key in the registry of Windows NT versions can have an associated security descriptor. The security descriptor contains an access control list (ACL) that describes which user groups or individual users are granted or denied access permissions. The set of registry permissions include 10 rights/permissions which can be explicitly allowed or denied to a user or a group of users. Registry permissions Permission Description Query Value The right to read the registry key value. Set Value The right to write a new value Create Subkey The right to create subkeys. Enumerate Subkeys Allow the enumeration of subkeys. Notify The right to request change notifications for registry keys or subkeys. Create Link Reserved by the operating system. Delete The right to delete a key. Write DACL The right to modify permissions of the container's DACL. Write Owner The right to modify the container's owner. Read Control The right to read the DACL. As with other securable objects in the operating system, individual access control entries (ACE) on the security descriptor can be explicit or inherited from a parent object.[42] Windows Resource Protection is a feature of Windows Vista and later versions of Windows that uses security to deny Administrators and the system WRITE access to some sensitive keys to protect the integrity of the system from malware and accidental modification.[43] Special ACEs on the security descriptor can also implement mandatory integrity control for the registry key and subkeys. A process running at a lower integrity level cannot write, change or delete a registry key/value, even if the account of the process has otherwise been granted access through the ACL. For instance, Internet Explorer running in Protected Mode can read medium and low integrity registry keys/values of the currently logged on user, but it can only modify low integrity keys.[44] Outside security, registry keys cannot be deleted or edited due to other causes. Registry keys containing NUL characters cannot be deleted with standard registry editors and require a special utility for deletion, such as RegDelNull.[45][46] Different editions of Windows have supported a number of different methods to back up and restore the registry over the years, some of which are now deprecated: System Restore can back up the registry and restore it as long as Windows is bootable, or from the Windows Recovery Environment (starting with Windows Vista). NTBackup can back up the registry as part of the System State and restore it. Automated System Recovery in Windows XP can also restore the registry. On Windows NT, the Last Known Good Configuration option in startup menu relinks the HKLM\SYSTEM\CurrentControlSet key, which stores hardware and device driver information. Windows 98 and Windows ME include command line (Scanreg.exe) and GUI (Scanregw.exe) registry checker tools to check and fix the integrity of the registry, create up to five automatic regular backups by default and restore them manually or whenever corruption is detected.[47] The registry checker tool backs up the registry, by default, to %Windir%\Sysbckup Scanreg.exe can also run from MS-DOS.[48] The Windows 95 CD-ROM included an Emergency Recovery Utility (ERU.exe) and a Configuration Backup Tool (Cfgback.exe) to back up and restore the registry. Additionally Windows 95 backs up the registry to the files system.da0 and user.da0 on every successful boot. Windows NT 4.0 included RDISK.EXE, a utility to back up and restore the entire registry.[49] Windows 2000 Resource Kit contained an unsupported pair of utilities called Regback.exe and RegRest.exe for backup and recovery of the registry.[50] Periodic automatic backups of the registry are now disabled by default on Windows 10 May 2019 Update (version 1903). Microsoft recommends System Restore be used instead.[51] Windows 2000 and later versions of Windows use Group Policy to enforce registry settings through a registry-specific client extension in the Group Policy processing engine.[52] Policy may be applied locally to a single computer using gpedit.msc or to multiple users and computers in a domain using gpmc.msc. With Windows 95, Windows 98, Windows ME and Windows NT 4.0, administrators can use a special file to be merged into the registry, called a policy file (POLICY.POL). The policy file allows administrators to prevent non-administrator users from changing registry settings like, for instance, the security level of Internet Explorer and the desktop background wallpaper. The policy file is primarily used in a business with a large number of computers where the business needs to be protected from rogue or careless users. The default extension for the policy file is .POL. The policy file filters the settings it enforces by user and by group (a "group" is a defined set of users). To do that the policy file merges into the registry, preventing users from circumventing it by simply changing back the settings. The policy file is usually distributed through a LAN, but can be placed on the local computer. The policy file is created by a free tool by Microsoft that goes by the filename poledit.exe for Windows 95/Windows 98 and with a computer management module for Windows NT. The editor requires administrative permissions to be run on systems that uses permissions. The editor can also directly change the current registry settings of the local computer and if the remote registry service is installed and started on another computer it can also change the registry on that computer. The policy editor loads the settings it can change from .ADM files, of which one is included, that contains the settings the Windows shell provides. The .ADM file is plain text and supports easy localisation by allowing all the strings to be stored in one place. Windows NT kernels support redirection of INI file-related APIs into a virtual file in a registry location such as HKEY_CURRENT_USER using a feature called "InifileMapping".[53] This functionality was introduced to allow legacy applications written for 16-bit versions of Windows to be able to run under Windows NT platforms on which the System folder is no longer considered an appropriate location for user-specific data or configuration. Non-compliant 32-bit applications can also be redirected in this manner, even though the feature was originally intended for 16-bit applications. Windows Vista introduced limited registry virtualization, whereby poorly written applications that do not respect the principle of least privilege and instead try to write user data to a read-only system location (such as the HKEY_LOCAL_MACHINE hive), are silently redirected to a more appropriate location, without changing the application itself. Similarly, application virtualization redirects all of an application's invalid registry operations to a location such as a file. Used together with file virtualization, this allows applications to run on a machine without being installed on it. Low integrity processes may also use registry virtualization. For example, Internet Explorer 7 or 8 running in "Protected Mode" on Windows Vista and above will automatically redirect registry writes by ActiveX controls to a sandboxed location in order to frustrate some classes of security exploits. The Application Compatibility Toolkit[54] provides shims that can transparently redirect HKEY_LOCAL_MACHINE or HKEY_CLASSES_ROOT Registry operations to HKEY_CURRENT_USER to address "LUA" bugs that cause applications not to work for users with insufficient rights. Critics labeled the registry in Windows 95 a single point of failure, because re-installation of the operating system was required if the registry became corrupt. However, Windows NT uses transaction logs to protect against corruption during updates. Current versions of Windows use two levels of log files to ensure integrity even in the case of power failure or similar catastrophic events during database updates.[55] Even in the case of a non-recoverable error, Windows can repair or re-initialize damaged registry entries during system boot.[55] This section needs additional citations for verification. Please help improve this article by adding citations to reliable sources in this section. Unsourced material may be challenged and removed. (November 2010) (Learn how and when to remove this message) In Windows, use of the registry for storing program data is a matter of developer's discretion. Microsoft provides programming interfaces for storing data in XML files (via MSXML) or database files (via SQL Server Compact) which developers can use instead. Developers are also free to use non-Microsoft alternatives or develop their own proprietary data stores. In contrast to Windows Registry's binary-based database model, some other operating systems use separate plain-text files for daemon and application configuration, but group these configurations together for ease of management. In Unix-like operating systems (including Linux) that follow the Filesystem Hierarchy Standard, system-wide configuration files (information similar to what would appear in HKEY_LOCAL_MACHINE on Windows) are traditionally stored in files in /etc/ and its subdirectories, or sometimes in /usr/local/etc/. Per-user information (information that would be roughly equivalent to that in HKEY_CURRENT_USER) is stored in hidden directories and files (that start with a period/full stop) within the user's home directory. However XDG-compliant applications should refer to the environment variables defined in the Base Directory specification.[56] In macOS, system-wide configuration files are typically stored in the /Library/ folder, whereas per-user configuration files are stored in the corresponding ~/Library/ folder in the user's home directory, and configuration files set by the system are in /System/Library/. Within these respective directories, an application typically stores a property list file in the Preferences/ sub-directory. RISC OS (not to be confused with MIPS RISC/os) uses directories for configuration data, which allows applications to be copied into application directories, as opposed to the separate installation process that typifies Windows applications; this approach is also used on the ROX Desktop for Linux.[57] This directory-based configuration also makes it possible to use different versions of the same application, since the configuration is done "on the fly".[58] If one wishes to remove the application, it is possible to simply delete the folder belonging to the application.[59][60] This will often not remove configuration settings which are stored independently from the application, usually within the computer's !Boot structure, in !Boot.Choices or potentially anywhere on a network fileserver. It is possible to copy installed programs between computers running RISC OS by copying the application directories belonging to the programs, however some programs may require re-installing, e.g. when shared files are placed outside an application directory.[58] IBM AIX (a Unix variant) uses a registry component called Object Data Manager (ODM). The ODM is used to store information about system and device configuration. An extensive set of tools and utilities provides users with means of extending, checking, correcting the ODM database. The ODM stores its information in several files, default location is /etc/objrepos. The GNOME desktop environment uses a registry-like interface called dconf for storing configuration settings for the desktop and applications. The Elektra Initiative provides alternative back-ends for various different text configuration files. While not an operating system, the Wine compatibility layer, which allows Windows software to run on a Unix-like system, also employs a Windows-like registry as text files in the WINEPREFIX folder: system.reg (HKEY_LOCAL_MACHINE), user.reg (HKEY_CURRENT_USER) and userdef.reg.[61] Registry cleaner Application virtualization LogParser – SQL-like querying of various types of log files List of Shell Icon Overlay Identifiers Ransomware attack that uses Registry ^ When applications fail to execute because they request more privileges than they require (and are denied those privileges), this is known as a limited user application (LUA) bug. ^ Esposito, Dino (November 2000). "Windows 2000 Registry: Latest Features and APIs Provide the Power to Customize and Extend Your Apps". MSDN Magazine. Microsoft. Archived from the original on April 15, 2003. Retrieved July 19, 2007. ^ a b c "The System Registry". ^ "Windows 95 Architecture Components". www.microsoft.com. Archived from the original on February 7, 2008. Retrieved April 29, 2008. The following table shows other difficulties or limitations caused by using .INI files that are overcome by using the Registry. ^ Hipson 2002, p. 5, 41–43. ^ Richter, Jeffrey; Nasarre, Christophe (2008). Windows Via C/C++ (Fifth ed.). Microsoft Press. ISBN 9780735642461. Retrieved August 28, 2021. ^ Raymond Chen, "Why do registry keys have a default value?" ^ Hipson 2002, pp. 207, 513–514. ^ Hipson 2002, pp. 520–521. ^ Hipson 2002, p. 7. ^ "Designed for Windows XP Application Specification". Microsoft. August 20, 2002. Retrieved April 8, 2009. ^ "HKEY_LOCAL_MACHINE". Gautam. 2009. Retrieved April 8, 2009. ^ "Registry Keys Affected by WOW64 (Windows)". Msdn.microsoft.com. Retrieved April 10, 2014. ^ "Description of the Microsoft Windows registry". Retrieved September 25, 2008. ^ "HKEY_CURRENT_USER". Microsoft. 2009. Retrieved April 8, 2009. ^ "Description of the HKEY_DYN_DATA Registry Key in Windows 95, Windows 98, and Windows 98 SE". support.microsoft.com. ^ "A Closer Look at HKEY_DYN_DATA". rinet.ru. Archived from the original on May 9, 2008. ^ "Registry hives". Retrieved July 19, 2007. ^ Chen, Raymond (August 8, 2011). "Why is a registry file called a "hive"?". The Old New Thing. Retrieved July 29, 2011. ^ "Overview of the Windows NT Registry". Retrieved December 2, 2011. ^ "Inside the Registry". Retrieved December 28, 2007. ^ a b Norris, Peter (February 2009). "The Internal Structure of the Windows Registry" (PDF). Cranfield University. Archived from the original (PDF) on May 29, 2009. ^ "Incorrect Icons Displayed for .ico Files". November 15, 2009. Retrieved March 31, 2012. ^ "How to Completely Uninstall / Remove a Software Program in Windows without using 3rd Party Software? - AskVG". www.askvg.com. August 26, 2011. ^ "You may receive a "STOP 0x00000035 NO_MORE_IRP_STACK_LOCATIONS" error message when you try to log on to a domain". October 9, 2011. Retrieved March 31, 2012. This page tells the user to edit the registry when resolving the issue. ^ key renaming is implemented as removal and add while retaining subkeys/values, as the underlying APIs do not support the rename function directly ^ a b c d e "How to add, modify, or delete registry subkeys and values by using a .reg file". support.microsoft.com. ^ "Applying Group Policy". Microsoft. ^ a b c Payette, Bruce; Siddaway, Richard (2018). Windows PowerShell in Action (Third ed.). Manning Publications. pp. 7–8, 24, 608, 708–710. ISBN 9781633430297. Retrieved August 28, 2021. ^ Warner, Timothy L. (May 2015). Windows PowerShell in 24 Hours, Sams Teach Yourself. Sams Publishing. p. 19, 211. ISBN 9780134049359. Retrieved August 28, 2021. ^ "Reading and Writing Registry Values with Visual Basic". Retrieved July 19, 2007. ^ "REG command in Windows XP". Retrieved July 19, 2007. ^ "registry manual page – Tcl Bundled Packages". www.tcl.tk. Retrieved December 14, 2017. ^ "Offline Registry Library". Retrieved June 4, 2014. ^ "DllInstall Function". Microsoft. March 7, 2012. Retrieved March 22, 2012. ^ "Regsvr32". Microsoft. Retrieved March 22, 2012. ^ "How to: Register Automation Servers". Microsoft. Retrieved March 22, 2012. ^ "How to re-register PowerPoint 2000, PowerPoint 2003, PowerPoint 2007 and PowerPoint 2010". Microsoft. January 2012. Retrieved March 22, 2012. ^ "RegNotifyChangeKeyValue function". Microsoft. ^ "Registering for Notifications". Microsoft. ^ "RegConnectRegistry function". Microsoft. ^ "How to Manage Remote Access to the Registry". Microsoft. ^ Gibson, Darril (June 28, 2011). "Chapter 4: Securing Access with Permissions". Microsoft Windows security : essentials. Indianapolis, Ind.: Wiley. ISBN 978-1-118-01684-8. ^ "Application Compatibility: Windows Resource Protection (WRP)". Microsoft. Retrieved August 8, 2012. ^ Marc Silbey, Peter Brundrett. "Understanding and Working in Protected Mode Internet Explorer". Retrieved August 8, 2012. ^ "RegDelNull v1.1". November 1, 2006. Retrieved August 8, 2012. ^ "Unable to delete certain registry keys – Error while deleting key". March 23, 2010. Retrieved August 8, 2012. Microsoft Support page. ^ "Description of the Windows Registry Checker Tool (Scanreg.exe)". ^ "Command-Line Switches for the Registry Checker Tool". ^ "How To Backup, Edit, and Restore the Registry in Windows NT 4.0". support.microsoft.com. ^ "Technical Reference to the Registry: Related Resources". Microsoft. Retrieved September 9, 2011. ^ Whitwam, Ryan (July 2019). "Microsoft Kills Automatic Registry Backups in Windows 10". ExtremeTech. Retrieved July 1, 2019. ^ "How Core Group Policy Works". Microsoft. September 2, 2009. Retrieved August 13, 2012. ^ "Chapter 26 – Initialization Files and the Registry". Microsoft. Retrieved March 3, 2008. ^ "Microsoft Application Compatibility Toolkit 5.0". Microsoft. Retrieved July 26, 2008. ^ a b Ionescu, Mark Russinovich, David A. Solomon, Alex (2012). "Registry Internals". Windows internals (6th ed.). Redmond, Wash.: Microsoft Press. ISBN 978-0-7356-4873-9.{{cite book}}: CS1 maint: multiple names: authors list (link) ^ "XDG Base Directory Specification". standards.freedesktop.org. ^ "Application directories". Archived from the original on May 27, 2012. Retrieved May 17, 2012. ^ a b "Case Studies Of The Top 132 Annoyances With Operating Systems Other Than RISC OS". Retrieved April 3, 2012. Page from the riscos.com website. Mentioned in points 82 and 104. ^ "RISC OS tour". Retrieved July 19, 2007. ^ "The RISC OS Products Directory". November 2, 2006. Archived from the original on February 19, 2007. Retrieved April 1, 2012. ^ 3.2. Using the Registry and Regedit (Wine User Guide) Hipson, Peter (2002). Mastering Windows XP Registry. Wiley. ISBN 0-7821-2987-0. Retrieved August 28, 2021. Russinovich, Mark E.; Solomon, David A. (2005). Microsoft Windows Internals (Fourth ed.). Microsoft Press. pp. 183–236. ISBN 978-0-7356-1917-3. Wikibooks has a book on the topic of: Windows registry hacks Windows Registry info & reference in the MSDN Library Retrieved from " If you are wondering how to clear Remote Desktop Connection history, the first step you should be able to connect your computer to another computer via Remote Desktop Connection. To be honest with you, a remote desktop is used to connect to another computer each time, Windows always keeps a list of each connection in the history list. Thus, it is a good choice for you to delete this history. In this text, we show the solution of deleting history entry for Remote Desktop Connection in Windows 10. Steps to delete history entry for Remote Desktop Connection in Windows 10 Step 1: Open Registry Editor in Windows 10. Step 2: To delete the Windows 10 Remote Desktop Connection history, you need to operate the registry entry firstly, for example, carry out following the registry command: HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default Note: By the way, using registry entry is so dangerous and difficult, because a small error can negatively impact the entire system. Therefore, it is necessary to be careful when using the registry. If there is a system error, you can use the registry repair tool, it will automatically repair the registry problem. Step 3: After finishing the above steps, you will see a list of registry string named MRU0, 1, 2, 3... Step 4: Right-click this MRU entry and choose the Delete button from the context menu. Tips: You can close the Registry Editor window and restart your PC. After restarting the system, you will find that the latest remote desktop connection history does not appear anymore. Related Articles