NAME

virt−win−reg − Export and merge Windows Registry entries from a Windows guest

SYNOPSIS

```
virt−win−reg domname 'HKLM\Path\To\Subkey'

virt−win−reg domname 'HKLM\Path\To\Subkey' name

virt−win−reg domname 'HKLM\Path\To\Subkey' @

virt−win−reg −−merge domname [input.reg ...]

virt−win−reg [−−options] disk.img ... # instead of domname
```

WARNING

You must *not* use `virt−win−reg` with the *−−merge* option on live virtual machines. If you do this, you *will* get irreversible disk corruption in the VM. `virt−win−reg`tries to stop you from doing this, b ut doesn't catch all cases.

Modifying the Windows Registry is an inherently risky operation. The format is deliberately obscure and undocumented, and Registry changes can leave the system unbootable. Therefore when using the *−−merge* option, make sure you have a reliable backup first.

DESCRIPTION

This program can export and merge Windows Registry entries from a Windows guest.

The first parameter is the libvirt guest name or the raw disk image of a Windows guest.

If *−−merge* is *not* specified, then the chosen registry key is displayed/exported (recursively). For example:

```
$ virt−win−reg Windows7 'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft'
```

You can also display single values from within registry keys, for example:

```
$ cvkey='HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion'
$ virt−win−reg Windows7 $cvkey ProductName
Windows 7 Enterprise
```

With *−−merge*, you can merge a textual regedit file into the Windows Registry:

```
$ virt−win−reg −−merge Windows7 changes.reg
```

NOTE

This program is only meant for simple access to the registry. If you want to do complicated things with the registry, we suggest you download the Registry hive files from the guest using **libguestfs**(3) or **guestfish**(1) and access them locally, eg. using **hivex**(3), **hivexsh**(1) or **hivexregedit**(1).

OPTIONS

−−help
Display brief help.

−−version
Display version number and exit.

−−debug
Enable debugging messages.

−c URI
−−connect URI
If using libvirt, connect to the given *URI*. If omitted, then we connect to the default libvirt hypervisor.

If you specify guest block devices directly, then libvirt is not used at all.

--format raw
Specify the format of disk images given on the command line. If this is omitted then the format is autodetected from the content of the disk image.

If disk images are requested from libvirt, then this program asks libvirt for this information. In this case, the value of the format parameter is ignored.

If working with untrusted raw-format guest disk images, you should ensure the format is always specified.

--merge
In merge mode, this merges a textual regedit file into the Windows Registry of the virtual machine. If this flag is *not* given then virt-win-reg displays or exports Registry entries instead.

Note that *--merge* is *unsafe* to use on live virtual machines, and will result in disk corruption. However exporting (without this flag) is always safe.

--encoding UTF-16LE|ASCII
When merging (only), you may need to specify the encoding for strings to be used in the hive file. This is explained in detail in ''ENCODING STRINGS'' in **Win::Hivex::Regedit** (3).

The default is to use UTF-16LE, which should work with recent versions of Windows.

--unsafe-printable-strings
When exporting (only), assume strings are UTF-16LE and print them as strings instead of hex sequences. Remove the final zero codepoint from strings if present.

This is unsafe and does not preserve the fidelity of strings in the original Registry for various reasons:

- Assumes the original encoding is UTF-16LE. ASCII strings and strings in other encodings will be corrupted by this transformation.
- Assumes that everything which has type 1 or 2 is really a string and that everything else is not a string, but the type field in real Registries is not reliable.
- Loses information about whether a zero codepoint followed the string in the Registry or not.

This all happens because the Registry itself contains no information about how strings are encoded (see ''ENCODING STRINGS'' in **Win::Hivex::Regedit** (3)).

You should only use this option for quick hacking and debugging of the Registry contents, and *never* use it if the output is going to be passed into another program or stored in another Registry.

SUPPORTED SYSTEMS

The program currently supports Windows NT-derived guests starting with Windows XP through to at least Windows 8.

The following Registry keys are supported:

`HKEY_LOCAL_MACHINE\SAM`
`HKEY_LOCAL_MACHINE\SECURITY`
`HKEY_LOCAL_MACHINE\SOFTWARE`
`HKEY_LOCAL_MACHINE\SYSTEM`
`HKEY_USERS\.DEFAULT`
`HKEY_USERS\`*`SID`*
where *SID* is a Windows User SID (eg. `S-1-5-18`).

`HKEY_USERS\`*`username`*
where *username* is a local user name (this is a libguestfs extension).

You can use `HKLM` as a shorthand for `HKEY_LOCAL_MACHINE`, and `HKU` for `HKEY_USERS`.

The literal keys `HKEY_USERS\$SID` and `HKEY_CURRENT_USER` are not supported (there is no ''current user'').

WINDOWS 8

Windows 8 ‘‘fast startup’’ can prevent virt-win-reg from being able to edit the Registry. See ‘‘WINDOWS HIBERNATION AND WINDOWS 8 FAST STARTUP’’ in **guestfs**(3).

ENCODING

`virt-win-reg` expects that regedit files have already been reencoded in the local encoding. Usually on Linux hosts, this means UTF-8 with Unix-style line endings. Since Windows regedit files are often in UTF-16LE with Windows-style line endings, you may need to reencode the whole file before or after processing.

To reencode a file from Windows format to Linux (before processing it with the *--merge* option), you would do something like this:

```
iconv -f utf-16le -t utf-8 < win.reg | dos2unix > linux.reg
```

To go in the opposite direction, after exporting and before sending the file to a Windows user, do something like this:

```
unix2dos linux.reg | iconv -f utf-8 -t utf-16le > win.reg
```

For more information about encoding, see **Win::Hivex::Regedit**(3).

If you are unsure about the current encoding, use the **file**(1) command. Recent versions of Windows regedit.exe produce a UTF-16LE file with Windows-style (CRLF) line endings, like this:

```
$ file software.reg
software.reg: Little-endian UTF-16 Unicode text, with very long lines,
with CRLF line terminators
```

This file would need conversion before you could *--merge* it.

CurrentControlSet etc.

Registry keys like `CurrentControlSet` don’t really exist in the Windows Registry at the level of the hive file, and therefore you cannot modify these.

`CurrentControlSet` is usually an alias for `ControlSet001`. In some circumstances it might refer to another control set. The way to find out is to look at the `HKLM\SYSTEM\Select` key:

```
# virt-win-reg WindowsGuest 'HKLM\SYSTEM\Select'
[HKEY_LOCAL_MACHINE\SYSTEM\Select]
"Current"=dword:00000001
"Default"=dword:00000001
"Failed"=dword:00000000
"LastKnownGood"=dword:00000002
```

‘‘Current’’ is the one which Windows will choose when it boots.

Similarly, other `Current...` keys in the path may need to be replaced.

DELETING REGISTRY KEYS AND VALUES

To delete a whole registry key, use the syntax:

```
[-HKEY_LOCAL_MACHINE\Foo]
```

To delete a single value within a key, use the syntax:

```
[HKEY_LOCAL_MACHINE\Foo]
"Value"=-
```

WINDOWS TIPS

Note that some of these tips modify the guest disk image. The guest *must* be shut off, else you will get disk corruption.

RUNNING A BATCH SCRIPT WHEN A USER LOGS IN

Prepare a DOS batch script, VBScript or executable. Upload this using**guestfish**(1). F or this example the script is called `test.bat` and it is uploaded into `C:\`:

```
guestfish -i -d WindowsGuest upload test.bat /test.bat
```

Prepare a regedit file containing the registry change:

```
cat > test.reg <<'EOF'
[HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce]
"Test"="c:\\test.bat"
EOF
```

In this example we use the key `RunOnce` which means that the script will run precisely once when the first user logs in. If you want it to run every time a user logs in, replace `RunOnce` with `Run`.

Now update the registry:

```
virt-win-reg --merge WindowsGuest test.reg
```

INSTALLING A SERVICE

This section assumes you are familiar with Windows services, and you either have a program which handles the Windows Service Control Protocol directly or you want to run any program using a service wrapper like SrvAny or the free RHSrvAny.

First upload the program and optionally the service wrapper. In this case the test program is called `test.exe` and we are using the RHSrvAny wrapper:

```
guestfish -i -d WindowsGuest <<EOF
  upload rhsrvany.exe /rhsrvany.exe
  upload test.exe /test.exe
EOF
```

Prepare a regedit file containing the registry changes. In this example, the first registry change is needed for the service itself or the service wrapper (if used). The second registry change is only needed because I am using the RHSrvAny service wrapper.

```
cat > service.reg <<'EOF'
[HKLM\SYSTEM\ControlSet001\services\RHSrvAny]
"Type"=dword:00000010
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"="c:\\rhsrvany.exe"
"DisplayName"="RHSrvAny"
"ObjectName"="NetworkService"

[HKLM\SYSTEM\ControlSet001\services\RHSrvAny\Parameters]
"CommandLine"="c:\\test.exe"
"PWD"="c:\\Temp"
EOF
```

Notes:

- For use of `ControlSet001` see the section above in this manual page. You may need to adjust this according to the control set that is in use by the guest.
- `"ObjectName"` controls the privileges that the service will have. An alternative is `"ObjectName"="LocalSystem"` which would be the most privileged account.
- For the meaning of the magic numbers, see this Microsoft KB article: http://support.microsoft.com/kb/103000.

Update the registry:

```
virt-win-reg --merge WindowsGuest service.reg
```

SHELL QUOTING

Be careful when passing parameters containing \ (backslash) in the shell. Usually you will have to use 'single quotes' or double backslashes (but not both) to protect them from the shell.

Paths and value names are case-insensitive.

SEE ALSO

hivex (3), **hivexsh** (1), **hivexregedit** (1), **guestfs** (3), **guestfish** (1), **virt-cat** (1), **virt-tail** (1), **Sys::Guestfs** (3), **Win::Hivex** (3), **Win::Hivex::Regedit** (3), **Sys::Virt** (3), http://libguestfs.org/.

AUTHOR

Richard W.M. Jones http://people.redhat.com/~rjones/

COPYRIGHT

LICENSE

BUGS

To get a list of bugs against libguestfs, use this link: https://bugzilla.redhat.com/buglist.cgi?component=libguestfs&product=Virtualization+Tools

To report a new bug against libguestfs, use this link: https://bugzilla.redhat.com/enter_bug.cgi?component=libguestfs&product=Virtualization+Tools

When reporting a bug, please supply:

- The version of libguestfs.
- Where you got libguestfs (eg. which Linux distro, compiled from source, etc)
- Describe the bug accurately and give a way to reproduce it.
- Run **libguestfs-test-tool** (1) and paste the **complete, unedited** output into the bug report.