NAME

Win::Hivex::Regedit − Helper for reading and writing regedit format files

SYNOPSIS

```
use Win::Hivex;
use Win::Hivex::Regedit qw(reg_import reg_export);

$h = Win::Hivex->open ('SOFTWARE', write => 1);

open FILE, "updates.reg";
reg_import (\*FILE, $h);
$h->commit (undef);

reg_export ($h, "\\Microsoft\\Windows NT\\CurrentVersion", \*OUTFILE,
   prefix => "HKEY_LOCAL_MACHINE\\SOFTWARE");
```

DESCRIPTION

Win::Hivex::Regedit is a helper library for reading and writing the Windows regedit (or `.REG`) file format. This is the textual format that is commonly used on Windows for distributing groups of Windows Registry changes, and this format is read and written by the proprietary `reg.exe` and `regedit.exe` programs supplied with Windows. It is*not* the same as the binary ‘ ‘hive’’ format which the hivex library itself can read and write. Note that the regedit format is not well-specified, and hence deviations can occur between what the Windows program can read/write and what we can read/write. (Please file bugs for any deviations found).

Win::Hivex::Regedit is the low-level Perl library. There is also a command line tool for combining hive files and reg files (**hivexregedit**(1)). If you have a Windows virtual machine that you need to merge regedit-format changes into, use the high-level **virt−win−reg**(1) tool (part of libguestfs tools).

FUNCTIONS

reg_import

```
reg_import ($fh, ($h|$map), [encoding => "UTF−16LE"]);
```

This function imports the registry keys from file handle `$fh` either into the hive `$h` or via a map function.

The hive handle `$h` must have been opened for writing, ie. using the `write => 1` flag to `Win::Hivex->open`.

In the binary hive file, the first part of the key name (eg. `HKEY_LOCAL_MACHINE\SOFTWARE`) is not stored. You just have to know (somehow) that this maps to the `SOFTWARE` hive. Therefore if you are given a file containing a mixture of keys that have to be added to different hives, you have to have a way to map these to the hive handles. This is outside the scope of the hivex library, but if the second argument is a CODEREF (ie. reference to a function) then this `$map` function is called on each key name:

```
map ($keyname)
==> ($h, $keyname)
```

As shown, the function should return a pair, hive handle, and the true key name (with the prefix stripped off). For example:

```
sub map {
  if ($_[0] =~ /^HKEY_LOCAL_MACHINE\\SOFTWARE(.*)/i) {
    return ($software_h, $1);
  } else ...
}
```

`encoding` is the encoding used by default for strings. If not specified, this defaults to `"UTF−16LE"`, however we highly advise you to specify it. See ‘‘ENCODING STRINGS’’ below.

As with the regedit program, we merge the new registry keys with existing ones, and new node values with old ones. You can use the − (minus) character to delete individual keys and values. This is explained in detail in the Wikipedia page on the Windows Registry.

Remember you need to call `$h->commit (undef)` on the hivex handle before any changes are written to the hive file. See ''WRITING TO HIVE FILES'' in **hivex**(3).

reg_export

```
reg_export ($h, $key, $fh,
            [prefix => $prefix],
            [unsafe_printable_strings => 1]);
```

This function exports the registry keys starting at the root `$key` and recursively downwards into the file handle `$fh`.

`$key` is a case-insensitive path of the node to start from, relative to the root of the hive. It is an error if this path does not exist. Path elements should be separated by backslash characters.

`$prefix` is prefixed to each key name. The usual use for this is to make key names appear as they would on Windows. For example the key `\Foo` in the SOFTWARE Registry, with `$prefix` `HKEY_LOCAL_MACHINE\SOFTWARE`, would be written as:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Foo]
"Key 1"=...
"Key 2"=...
```

If `unsafe_printable_strings` is not given or is false, then the output is written as pure 7 bit ASCII, with line endings which are the default for the local host. Strings are always encoded as hex bytes. This is safe because it preserves the original content and encoding of strings. See ''ENCODING STRINGS'' below.

If `unsafe_printable_strings` is true, then strings are assumed to be UTF-16LE and are converted to UTF-8 for output. The final zero codepoint in the string is removed if there is one. This is unsafe because it does not preserve the fidelity of the strings in the Registry and because the content type of strings is not always UTF-16LE. However it is useful if you just want to display strings for quick hacking and debugging.

You may need to convert the file's encoding using **iconv**(1) and line endings using **unix2dos**(1) if sending to a Windows user.

Nodes and keys are sorted alphabetically in the output.

This function does *not* print a header. The real regedit program will print a header like:

```
Windows Registry Editor Version 5.00
```

followed by a blank line. (Other headers are possible, see the Wikipedia page on the Windows Registry). If you want a header, you need to write it out yourself.

reg_export_node

```
reg_export_node ($h, $node, $fh, ...);
```

This is exactly the same as ''reg_export'' except that instead of specifying the path to a key as a string, you pass a hivex library `$node` handle.

ENCODING STRINGS

The situation with encoding strings in the Registry on Windows is very confused. There are two main encodings that you would find in the binary (hive) file, 7 bit ASCII and UTF-16LE. (Other encodings are possible, it's also possible to have arbitrary binary data incorrectly marked with a string type).

The hive file itself doesn't contain any indication of string encoding. Windows probably guesses the encoding.

We think that regedit probably either guesses which encoding to use based on the file encoding, or else has different defaults for different versions of Windows. Neither choice is appropriate for a tool used in a real operating system.

When using ''reg_import'', you should specify the default encoding for strings using the `encoding` parameter. If not specified, it defaults to UTF-16LE.

The file itself that is imported should be in the local encoding for files (usually UTF-8 on modern Linux

systems). This means if you receive a regedit file from a Windows system, you may sometimes have to reencode it:

```
iconv -f utf-16le -t utf-8 < input.reg | dos2unix > output.reg
```

When writing regedit files (''reg_export'') we bypass this madness completely. *All* strings (even pure ASCII) are written as hex bytes so there is no doubt about how they should be encoded when they are read back in.

COPYRIGHT

LICENSE

SEE ALSO

Win::Hivex (3), **hivexregedit** (1), **virt-win-reg** (1), **iconv** (1), **dos2unix** (1), **unix2dos** (1), **hivex** (3), **hivexsh** (1), <http://libguestfs.org>, **Sys::Guestfs** (3).