

ThinApp Virtual Registry

ThinApp 4.6

Operating systems and applications store a large number of settings in the system registry. The virtual operating system intercepts requests to open files and redirects them to the virtual file system. The virtual operating system intercepts requests to look up and store values in the registry and redirects them to the virtual registry.

This information includes the following topics:

- [“Virtual Registry Formats”](#) on page 1
- [“Changing the Registry with vregtool.exe”](#) on page 4

Virtual Registry Formats

By storing settings in the virtual registry, ThinApp ensures that settings required by the application are accessible without actually changing the system registry.

The virtual registry uses the following formats:

- Build

The setup capture process generates this format that includes Unicode text files, such as `HKEY_LOCAL_MACHINE.txt`.

- Embedded

The build process converts the build format data into the embedded format data. ThinApp stores registry data inside of the primary data container.

- Sandbox

ThinApp stores registry data in the sandbox. As the application performs registry write operations, ThinApp stores the differences from the embedded format in sandbox format. When you start the application, ThinApp overlays the sandbox format data on the embedded format data to restore application settings from the last shutdown.

When you build a package, the registry values and files exist in a read-only image inside the executable file that you distribute. At runtime, ThinApp maps the read-only image as a view that presents a read-only view of the registry to the application. As the application modifies the registry, ThinApp saves these changes in sandbox format with the `.tvr` extension. ThinApp maps these changes as a read-write view of the registry to the application. You can use the `vregtool.exe` utility to list and modify data contained in a `.tvr` file.

ThinApp Build Format

The build format reveals the appearance of registry data after the capture process.

ThinApp stores each captured registry hive in its own file with the `.txt` extension. For example, ThinApp generates the `HKEY_LOCAL_MACHINE.txt` file and the `HKEY_CURRENT_USER.txt` file.

ThinApp uses the following format to store data in each file.

```
<isolation_mode> <complete_path_to_registry_key>
Value=<name>
<enumerated_type_for_value>
```

For example, the following entry might exist for the `Textpad.exe` application in the `HKEY_LOCAL_MACHINE.txt` file.

```
isolation_full HKEY_LOCAL_MACHINE\Software\Classes\*\OpenWithList\TextPad.exe
Value=
REG_SZ=#00
```

The type can appear as one of the following entries:

- REG_NONE
- REG_SZ
- REG_EXPAND_SZ
- REG_BINARY
- REG_DWORD
- REG_DWORD_LITTLE_ENDIAN
- REG_DWORD_BIG_ENDIAN
- REG_MULTI_SZ
- REG_RESOURCE_LIST
- REG_FULL_RESOURCE_DESCRIPTOR
- REG_RESOURCE_REQUIREMENTS_LIST

If the type is string data, such as `REG_SZ`, `REG_EXPAND_SZ`, or `REG_MULTI_SZ`, `Value` appears as a sequence of characters terminated by a new line or carriage return character. You can escape any unprintable characters into two hex characters.

```
REG_SZ=This is a line#23 with a carriage return as part of the data#00
REG_MULTI_SZ=This is a line1#00This is line2#23#00
```

Because the pound (`#`) sign represents an escape character, all `#` characters must be escaped. ThinApp uses the percent (`%`) character for ThinApp macro expansion. This must be escaped when the value is not expanded.

Text Format for Virtual Registry

The text format for the virtual registry is similar to the `regedit.reg` format but supports macro expansion for string registry values names and value data.

The text format might resemble this format.

```
isolation_mode FULL_SUBKEY_NAME
Value=VALUE_NAME
VALUE_TYPE=VALUE_DATA
```

Another example of the format uses `MACRO_VALUE_DATA` for the `VALUE_TYPE`.

```
Value=VALUE_NAME
VALUE_TYPE~MACRO_VALUE_DATA
```

The `isolation_mode` can appear as `isolation_full`, `isolation_merged`, or `isolation_writecopy`.

The VALUE_TYPE specifies the value data registry type with one of the following entries:

- REG_SZ
- REG_EXPAND_SZ
- REG_BINARY
- REG_DWORD
- REG_DWORD_BIG_ENDIAN
- REG_LINK
- REG_MULTI_SZ
- REG_RESOURCE_LIST
- REG_FULL_RESOURCE_DESCRIPTOR
- REG_RESOURCE_REQUIREMENTS_LIST

If you need to represent a non-published registry type, you can specify a decimal base 10 number such as 453=#00.

The VALUE_DATA specifies data that is represented in different ways depending on whether the data type is a string or non-string type.

String types, such as REG_SZ, REG_MULTI_SZ, and REG_EXPAND_SZ, are values that ThinApp represents as escaped string values. You must use a unicode text file to represent non-ANSI characters. ThinApp uses unicode text files by default. The pound (#), new line, tab, carriage return, end-of-file, and NULL characters must be escaped if they exist in the original string.

Escaped characters begin with # and are followed by two hex characters to represent the ANSI value for the character. Non-ANSI unicode characters do not need to be escaped but can only exist in unicode text files.

This example represents a five character string value. The fifth character is a NULL (0) character.

REG_SZ=Both#00

This example represents a multi-string value that contains String1 and String2.

REG_MULTI_SZ=String1#00String2#00#00

This example represents a four byte binary value.

REG_DWORD=#27#c6#00#02

For binary values, ThinApp stores data as a string of escaped bytes. For DWORD values, ThinApp stores data in native x86 order (little endian) that does not require special processing for binary data.

The MACRO_VALUE_DATA specifies value data that requires macro expansion before the application uses it. For example, ThinApp expands %AppData% at runtime to the location of the user's Application Data directory. If the value data is not expanded, the application receives the literal %AppData% string when it queries this registry value. ThinApp supports macro-expanded data for string value types, such as REG_SZ, REG_EXPAND_SZ, and REG_MULTI_SZ.

This example represents the value of the Application Data folder on the computer where ThinApp runs.

REG_SZ~%AppData%

The length of this registry value changes depending on the computer.

This example represents the value of the actual %AppData% value. This value is always the literal %AppData% string regardless of the current shell folder location.

REG_SZ~#23AppData#23

Here is another example.

```
isolation_full HKEY_LOCAL_MACHINE\Software\Classes\CLSID\{047a9a40-657e-11d3-8d5b-00104b35e7ef}\InprocServer32
Value=ThreadingModel
REG_SZ=Both#00
Value=
REG_SZ~%SystemSystem%\mscoree.dll#00
```

Changing the Registry with vregtool.exe

The `vregtool.exe` utility compiles the virtual registry during the build process. You can use this utility to manipulate the `.tvr` files of the virtual registry.

The following command-line usage uses parameters that are not case sensitive.

```
vregtool regfile.tvr ImportDir InputDirectory
vregtool regfile.tvr ImportReg regedit.reg [-Merged|-WriteCopy|-Full] [-NoReplace][-NoMacros]
vregtool regfile.tvr ExportReg filename.reg [HKEY_LOCAL_MACHINE\Software]
vregtool regfile.tvr ExportTxt OutputDirectory [HKEY_LOCAL_MACHINE\Software]
vregtool regfile.tvr PrintKeys [HKEY_LOCAL_MACHINE\Software] [-ShowValues] [-ShowData] [-ExpandMacros]
vregtool regfile.tvr PrintStats
vregtool regfile.tvr SysCompare [HKEY_LOCAL_MACHINE\Software] [-Exact]
vregtool regfile.tvr DelSubkey HKEY_LOCAL_MACHINE\Software [-NoMark]
```

Performing Registry Operations

You can use the `vregtool.exe` utility to perform operations such as exporting registry data to `regedit` format, listing diagnostic information about a `.tvr` file, and deleting a registry subkey.

Importing Registry Data from ThinApp Registry Text Format

You can import registry data from the ThinApp registry text format.

```
VREGTOOL regfile.tvr ImportDir InputDirectory
```

`ImportDir` is a keyword.

The `regfile.tvr` is the data file in ThinApp virtual registry file format.

`InputDirectory` is the directory where ThinApp reads registry data.

This is an example of importing registry data from ThinApp registry text format.

```
vregtool c:\tmp\test.tvr ImportDir %ProjectDir%
```

Importing Registry Data from Regedit Format

You can import registry data from `Regedit` format.

```
VREGTOOL regfile.tvr ImportReg regedit.reg [-Merged|-WriteCopy|-Full] [-NoReplace][-NoMacros]
```

The `regfile.tvr` file is the data file in ThinApp virtual registry file format.

The `regedit.reg` file is the file to import. ThinApp adds imported entries to the specified `.tvr` file. This file can use REGEDIT 4.0 (ansi text) or 5.0 (unicode text) format.

The isolation mode options specify `Merged`, `WriteCopy`, or `Full` isolation mode for keys that do not already exist. If you do not specify an isolation mode, ThinApp assigns `WriteCopy` isolation mode.

The `NoReplace` option prevents ThinApp from replacing or modifying existing registry values. If you do not select this option and a `.tvr` file already contains a value specified in the `.reg` file, ThinApp overwrites this value with the value specified in the `.reg` file.

The NoMacros option prevents ThinApp from performing macro substitution for registry values that contain paths to short path name or shell folders. When you do not set this option, ThinApp replaces the values contained in the .tvr file with macro versions of paths. For example, if the .reg file contains a string registry value of C:\windows\system32\kernel32.dll, the .tvr file contains the %systemsystem%\kernel32.dll value. When the application requests the value of this registry key, it receives the C:\windows\system\kernel32.dll value when running on Windows 98, Windows ME, and Windows XP+. If the application runs on Windows NT or Windows 2000, it receives the C:\winnt\system32\kernel32.dll value.

In most cases, you can use macro substitution. You might disable this option and store a hard-coded path in the registry.

Exporting Registry Data to Regedit Format

You can export registry data to Regedit format.

```
VREGTOOL regfile.tvr ExportReg filename.reg [HKEY_LOCAL_MACHINE\Software]
```

The regfile.tvr file is the data file in ThinApp virtual registry file format.

The regedit.reg file is the .reg file to export to. The file is in REGEDIT 5.0 format (unicode text).

HKEY_LOCAL_MACHINE\Software specifies a registry subtree to export. If you do not specify a subtree, the vregtool.exe utility exports the entire contents of the .tvr file. If the specified registry subkey has a space in the name, specify the key with quotes. For example, specify "HKEY_LOCAL_MACHINE\Software\Key with space".

When you export registry data to .reg format, ThinApp can lose some information.

- Filename macros are expanded during the export process. If the filenames are not converted back to macros when you import, ThinApp might lose some information.
- Each registry subkey in .tvr format has a specified isolation mode. Because the .reg format does not detect isolation modes or metadata for subkeys, ThinApp loses this information.
- Registry values that cannot be represented in .reg format are lost. For example, a key that is REG_SZ cannot have more than one NULL character in .reg format. In this case, the registry value data is prematurely truncated in .reg format.

An example of when the .reg files cannot represent values accurately is a REG_SZ value that is not null-terminated.

To avoid the loss of information, you can export registry data to ThinApp registry text format and re-import the data into the .tvr file.

```
vregtool c:\tmp\test.tvr ExportReg c:\tmp\test.reg "HKEY_CURRENT_USER\Software\Adobe\Save For Web 3.0"
```

Listing all Registry Keys in a ThinApp .tvr File

You can list all registry keys in a .tvr file.

```
VREGTOOL regfile.tvr PrintKeys [HKEY_LOCAL_MACHINE\Software] [-ShowValues] [-ShowData] [-ExpandMacros]
```

The regfile.tvr file is the data file in ThinApp virtual registry file format.

HKEY_LOCAL_MACHINE\Software specifies a registry subtree to print. If you do not specify a subtree, the vregtool.exe utility prints the entire contents of the .tvr file. If the specified registry subkey has a space in the name, specify the key with quotes. For example, specify "HKEY_LOCAL_MACHINE\Software\Key with space".

The ShowValues option prints the names of the virtual values contained in virtual subkeys.

The ShowData option prints the data associated with each virtual value.

The ExpandMacros option expands macros contained in registry values and data before printing.

For example, this operation prints all the virtual registry keys contained in a .tvr file to the console.

```
vregtool c:\tmp\test.tvr PrintKeys "HKEY_CURRENT_USER\Software\Adobe\Save For Web 3.0"
```

Listing Diagnostic Information About a Thinapp.tvr File

You can list diagnostic information about a .tvr file.

```
VREGTOOL regfile.tvr PrintStats
```

The `regfile.tvr` file is the data file in ThinApp virtual registry file format.

You can use this option to troubleshoot issues with .tvr files.

```
vregtool c:\tmp\test.tvr PrintStats
```

Comparing Virtual Registry Information with Host Computer Registry Information

You can compare virtual registry information with the host computer registry information.

```
VREGTOOL regfile.tvr SysCompare [HKEY_LOCAL_MACHINE\Software] [-Exact]
```

The `regfile.tvr` file is the data file in ThinApp virtual registry file format.

`HKEY_LOCAL_MACHINE\Software` specifies the registry subkey to start the comparison. If you specify this information, the subkeys at this level and below this level are considered in the comparison.

The `-Exact` option specifies that the system comparison reports all differences between the virtual registry and system registry. If you do not specify this option, the `vregtool.exe` utility does not print the system registry keys that do not exist in the virtual registry if the virtual registry subkey is set to Merged or WriteCopy isolation mode. In this case, the `vregtool.exe` utility prints the differences for subkeys with Full isolation mode.

This example locates the differences between a working system registry and virtual registry file.

```
vregtool c:\tmp\test.tvr SysCompare "HKEY_CURRENT_USER\Software\Adobe"
```

Deleting a Registry Subkey

You can delete a registry subkey.

```
VREGTOOL regfile.tvr DelSubkey HKEY_LOCAL_MACHINE\Software [-NoMark]
```

The `regfile.tvr` file is the data file in ThinApp virtual registry file format.

`HKEY_LOCAL_MACHINE\Software` specifies the registry subkey to delete. If the specified registry subkey has a space in the name, specify the key with quotes. For example, specify `"HKEY_LOCAL_MACHINE\Software\Key with space"`.

The `NoMark` option specifies that ThinApp can delete the subkey if it exists, but not mark it as deleted. When a subkey is marked as deleted, applications running under ThinApp do not see the specified subkey even if it exists on the local system.

If you do not use the `-NoMark` option, the `vregtool.exe` utility marks the subkey as deleted regardless of whether it originally exists in the .tvr file.

```
vregtool c:\tmp\test.tvr DelSubkey "HKEY_CURRENT_USER\Software\Adobe\Save For Web 3.0"
```

If you have comments about this documentation, submit your feedback to: docfeedback@vmware.com

VMware, Inc. 3401 Hillview Ave., Palo Alto, CA 94304 www.vmware.com

Copyright © 2010 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Item: EN-000370-01
