

WinHelp unter Windows 10

Wer unter dem Microsoft-Betriebssystem *Windows 10* ältere Programme installiert hat, bei denen noch das 1990 eingeführte Dateiformat *WinHelp* integriert ist, kann die Hilfedateien nicht mehr aufrufen. Mit Einführung von *Vista* wurde auf eine Unterstützung verzichtet, aber immer wieder bis einschließlich *Windows 7* aus Kompatibilitätsgründen nachgeliefert. Danach sollte nur noch das 1997 eingeführte CHM-Dateiformat Verwendung finden. Mehr zu *WinHelp* ist in [1] zu finden.

Wer in einem solchen „älteren“ Programm die Hilfe aufruft, landet bei Verbindung zum Internet auf der Supportseite von Microsoft mit dem Hinweis, dass ein Fehler beim Öffnen der Hilfe in Windows-basierten Programmen aufgetreten ist bzw. die *Hilfe nicht unterstützt* wird. Das für die Anzeige der Hilfedatei erforderliche Programm *WinHlp32.exe* lässt sich jedoch nur bis zur Version *Windows 8.1* nachladen und installieren [2].

Im Internet findet man verschiedene Varianten zur Behebung dieses Problems. Diese reichen von kostenpflichtigen Konvertierungsprogrammen bis hin zur Erstellen von PDF-Dateien. All diese Lösungsansätze waren für mich unbefriedigend. In [3] ist aber einen Hinweis zur Behebung des Problems

zu finden: Es wird auf die Website von Komeil Bahmanpour hingewiesen [4], auf der sich ein Download für die schon betagte *winhelp32-windows-7-x86-x64-komeil.cab* befindet. Darin sind die Dateien *Install.cmd*, *winhelp32.exe* und *winhelp32.exe.mui* enthalten.

In dieser Form lässt sich aber die *Winhelp*-Datei nicht für *Windows 10* installieren. Sie ist nur bis *Windows 7* geeignet. Es ist deshalb eine Anpassung bei der Installationsroutine erforderlich. In [3] ist auch ein entsprechender Hinweis zu finden, wie die Installationsdatei zu modifizieren ist: Herunterladen der Datei von [4] und zwei Zeilen als Administrator einfügen.

Ich habe dazu beim 64-Bit-Windows die CAB-Datei in ein Verzeichnis unter *Programme (x86)* abgelegt, die darin enthaltenen drei Dateien entpackt und mithilfe des Windows-eigenen Editors die *Install.cmd* geöffnet. Die beiden neuen Zeilen sind nun unter *Settings* nach bereits bestehenden Anweisungen einzutragen. Bei *Settings* muss jetzt stehen:

```
:: Settings
set MuifileName=winhlp32.exe.mui
set ExeFileName=winhlp32.exe
set WindowsVersion=7
goto BypassVersionError
```

Dadurch wird der Installationsroutine jetzt die *Windows*-Version 7 vorgegeben. Es kann im Programm deshalb gleich zur Installation gesprungen und diese ausgeführt werden. Nach der Ergänzung der Datei *Install.cmd* das Abspeichern nicht vergessen.

Nun lässt sich *WinHelp32* mit der Datei *Install.cmd* (mit rechter Maustaste anklicken und als Administrator ausführen) problemlos unter *Windows 10* (32-Bit- und 64-Bit-Version) installieren. Wer sich die Arbeit erleichtern möchte, kann die Datei *WinHelpWin10.zip* aus dem Download-Bereich auf www.funkamateur.de herunterladen, in einem Verzeichnis (z. B. *WinHelp32*) unter *Programme (x86)* ablegen, entpacken, installieren und sofort loslegen.

Dr.-Ing. Klaus Wollert, DL7KWB
klaus.wollert@web.de

Literatur und Bezugsquellen

- [1] Wikipedia: Microsoft Help. https://de.wikipedia.org/wiki/Microsoft_Help
- [2] Microsoft Support: Fehler beim Öffnen der Hilfe in Windows-basierten Programmen. www.support.microsoft.com/de-de/kb/917607
- [3] Microsoft Community: Winhlp32 for Windows 10. http://answers.microsoft.com/en-us/Windows/Winhlp32_for_Windows_10 → Search: Winhlp32 for Windows 10
- [4] Bahmanpour, K.: Download Center: www.komeil.com/download/1230
- [5] Bahmanpour, K.: Windows Help Program (WinHlp32.exe) for Windows 8.1 and Windows 7 x64 and x86. www.komeil.com/blog/windows-help-program-winhelp-winhlp32-exe#comment-1520

Winziger Schriftfont auf Tablet-PC

Vor einiger Zeit kaufte ich einen Tablet-PC samt *Windows 10*. Nach der Installation des Programms für den Netzwerktester *FA-NWT01* ließ sich zwar das Programm starten, doch die Eingabefelder und Beschriftungen lagen teilweise übereinander. Auch ein via *Optionen* eingestellter kleinerer Schriftfont brachte keine nennenswerte Verbesserung. In den Einstellungen von *Windows* lässt sich auch die Auflösung ändern – bei meinem Tablet-PC ist sie standardmäßig 200 %. Das Verringern auf 100 % beseitigt zwar die Überlappungen, führt aber zu sehr kleinen und nur noch mit der Lupe lesbaren Schriftgrößen bei allen Programmen.

Bei [1] fand ich dann die Erklärung und die richtige Lösung für dieses Problem: Neuere Programme wurden für den Gebrauch auf Bildschirmen mit hoher Auflösung (DPI, engl. *dots per inch*) entwickelt. Sie sind DPI-bewusst (engl. *DPI aware*) und merken, wie hoch die Auflösung des Bildschirms ist, und greifen dann automatisch zu größeren Schriften. Älteren Programmen fehlt dieses DPI-Bewusstsein (engl. *DPI awareness*).

Soll so ein älteres Programm auch auf einem Tablet-PC genutzt werden, ist *Windows* mitzuteilen, selbst die Schriften dieser Anwendungen zu vergrößern. Das macht man mit je einer Manifest-Datei pro betroffenem Programm.

Das Vorgehen ist einfach, erfordert jedoch einen Eingriff in die *Registry* und ist daher mit Bedacht durchzuführen. Zuerst ist *Windows* zu melden, dass es grundsätzlich Manifest-Dateien berücksichtigen soll. Nach dem Drücken von *Windows*+*R* ist *regedit* einzugeben und die *Enter*-Taste zu betätigen. In der linken Spalte des sich dann öffnenden Fensters ist dem Pfad *HKEY_LOCAL_MACHINE* → *SOFTWARE* → *Microsoft* → *Windows* → *CurrentVersion* bis zum Eintrag *SideBySide* zu folgen.

Wenn beim Klicken auf den Zweignamen *SideBySide* rechts nicht der Eintrag *Prefer ExternalManifest* zu finden ist, muss er erstellt werden. Dazu mit der rechten Maustaste auf eine leere Stelle klicken und über *Neu* → *DWORD-Wert (32 Bit)* einen neuen Eintrag erzeugen. Als Name ist *Prefer ExternalManifest* einzugeben. Auch wenn dieser Eintrag schon vorhanden ist, steht

sein Wert in der Regel auf 0 – er ist auf 1 zu ändern.

Im zweiten Schritt wird z. B. die als Ergänzung auf www.funkamateur.de verfügbare Manifest-Datei in den Ordner des betroffenen Programms kopiert, in diesem Fall *Programme (x86)/AFU/WinNWT4*. Da das Programm hier *winnwt4.exe* heißt, muss die Manifest-Datei somit den Namen *winnwt4.exe.manifest* erhalten.

Das war es schon. Änderungen des Inhalts der nur ASCII-Zeichen enthaltenen Manifest-Datei sind in der Regel nicht erforderlich.

Diese kleinen Schriften sind keine Eigenart von *Windows 10*, denn auf einem Desktop-PC mit gleichem Betriebssystem ist die Darstellung von Hause aus korrekt. Dafür können Ihnen die zu kleinen Schriftfonten auch auf Tablet-PCs mit anderen *Windows*-Versionen begegnen.

Manifest-Dateien können auch dort nach dem hier gezeigten Verfahren eine verbesserte Schriftdarstellung erzielen.

Ingo Meyer, DK3RED
elektronik@funkamateur.de

Literatur

- [1] Antonielli, D.: Adobe App Scaling on High DPI Displays. www.danantonielli.com/adobe-app-scaling-on-high-dpi-displays-fix/