# MSAccess Anwendungen in Windows Vista

Jörg Ackermann
Oktober 2007

Seit einiger Zeit ist Windows Vista der auf dem Markt verfügbar. Der Nachfolger von Windows soll schneller, stabiler, sicherer und grafisch ansprechender sein. Neue Features und nicht unerhebliche Änderungen, vor allem die Funktionen zur Erhöhung der Sicherheit, erfordern von den Entwicklern einiges an Lernaufwand.
Obwohl sich viele derzeit noch zurückhalten, wird Vista das Betriebssystem der Zukunft sein. Entwickler werden gezwungen sein, auf Vista umzusteigen, wollen sie ihre berufliche Zukunft nicht gefährden.
Wie bei jeder Umstellung des Betriebssystems stellt sich natürlich auch für uns Accessler die bange Frage:

Laufen meine Access-Anwendungen, die bisher unter Windows XP klaglos ihren Dienst taten, unter Windows-Vista?

Die Antwort ist einfach: Im Prinzip ja.

Waren die Anwendungen so konzipiert, dass sie unter nichtadministrativen Berechtigungen laufen, wird es auch unter Vista keine (wesentlichen) Probleme geben. Wer allerdings unter XP mit Administrator- rechten gearbeitet hat, wird um Änderungen seiner Anwendung nicht herumkommen.

**Welche Probleme sind zu erwarten?**

- Probleme durch geänderte Dateisystem-Struktur
- Berechtigungsprobleme bei Dateisystem- und Registryzugriffen
- Probleme bei Installationen
- Probleme mit Verweisen
- Performanceprobleme

**Was kann ich dagegen tun?**

- Redesign bestehender Anwendungen
- Redesign von Installationspaketen
- Manuelle Eingriffe in die Registry

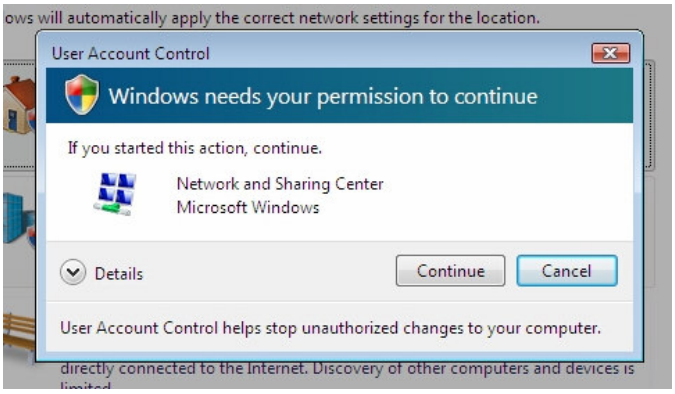
Bei den Vorgängerversionen von Microsoft Windows Vista sind die meisten Benutzerkonten Mitglieder der lokalen Gruppe Administratoren. Dies liegt - unabhängig von den Standard-Einstellungen bei der Installation des Betriebssystems - daran, dass für die Installation, Aktualisierung und Ausführung vieler Anwendungen sowie für einfachste administrative Aufgaben, wie z.B. das Einstellen der Systemzeit, entsprechende Rechte benötigt werden.

Windows Vista stellt einen einfachen und sicheren Mechanismus – User Account Control - zur Verfügung, mit dem Endbenutzerkonten mit Standardrechten ausgeführt werden können. Die Notwendigkeit von administrativen Rechten für das Ausführen von normalen Aufgaben wird beseitigt.

## Benutzerkontensteuerung - User Account Control (UAC)

**Was ist dieses UAC?**

Der erste Eindruck und zugleich das sichtbarste Feature von UAC ist die lästige Eingabeaufforderung zur Bestätigung erhöhter Rechte.




UAC ist also ein Versuch, uns in kürzester Zeit dazu zu bringen, dieses Feature abzuschalten?

UAC ist eine Sammlung von Features. Die Eingabeaufforderung zur Erhöhung der Rechte ist nur ein, wenn auch der sichtbarste Teil der Benutzerkontensteuerung.
Befassen wir uns zunächst etwas intensiver damit.

**Was also ist UAC wirklich?**

Bis einschließlich Server 2003 war es so, dass jeder Versuch ausschließlich mit Benutzerrechten sinnvoll produktiv zu arbeiten extrem anstrengend oder unmöglich war. Entwickler konnten praktisch gar nicht ohne Administratorrechte arbeiten.

UAC ist Version 2.0 des Versuchs, mehr Anwender in die Lage zu versetzen, nicht als Administrator zu arbeiten und mehr Entwickler dazu zu zwingen, Anwendungen zu schreiben, die mit Standard-Benutzerrechten ausgeführt werden können. Dies wiederum verringert die Anzahl der notwendigen Bestätigungen und soll dazu führen, dass Benutzer vor einer Bestätigung sorgfältiger nachdenken. Im Idealfall arbeiten die meisten Benutzer nicht als Administratoren und hinterfragen die Notwendigkeit der Rechterhöhung bei den Entwicklern. Ziele sind:

- Die Verringerung der Szenarios, die eine Erweiterung auf ein Administratorkonto benötigen.
- Einfachere Gestaltung der Rechterhöhung – möglichst Automatisierung – im Gegensatz zu den zuvor verwendeten umständlichen Methoden wie runas.exe oder die Abmeldung von einem Konto und die Anmeldung bei einem anderen.
- Bereitstellen einer gewissen Art von Isolierung für die Anwendungen, die mit Administratorberechtigungen ausgeführt werden, um sie vor denen ohne Administratorberechtigungen zu schützen.
- Anwendungen wann immer möglich erlauben, mit niedrigeren Berechtigungen ausgeführt zu werden.

Zwei dieser Ziele sollen die Benutzerfreundlichkeit verbessern, zwei sollen die Sicherheit erhöhen.

Erreicht wird dies durch verschiedene Verfahren, auf die ich nicht näher eingehen möchte. Interessierte sollten sich diese Technet-Artikel zum Thema anschauen.

*[1] Understanding and Configuring User Account Control in Windows Vista*
*http://technet.microsoft.com/en-us/windowsvista/aa905117.aspx*

*[2] Mark Russinovich*
*Die Benutzerkontensteuerung von Windows Vista*
*http://www.microsoft.com/technet/technetmag/issues/2007/06/UAC/default.aspx?loc=de*

*[3] Dr. Jesper M. Johansson*
*Sicherheit auf dem Prüfstand*
*http://www.microsoft.com/technet/technetmag/issues/2007/09/SecurityWatch/default.aspx?loc=de*

## Änderungen der Dateisystem-Struktur

Interessant für Access-Entwickler sind natürlich neben der Funktionsweise eher die Auswirkungen von UAC und damit die Anforderungen, die zukünftig an eine Anwendung zu stellen sind um mit diesen Rechten lauffähig zu sein.

Die standardmäßige Ausführung als Benutzer erfordert, sich an die vom Betriebssystem zur Verfügung gestellte Verzeichnisstruktur sowie die standardmäßigen Zugriffsrechte auf das Dateisystem und die Registry anzupassen. Ein guter Wegweiser ist folgendes Dokument:

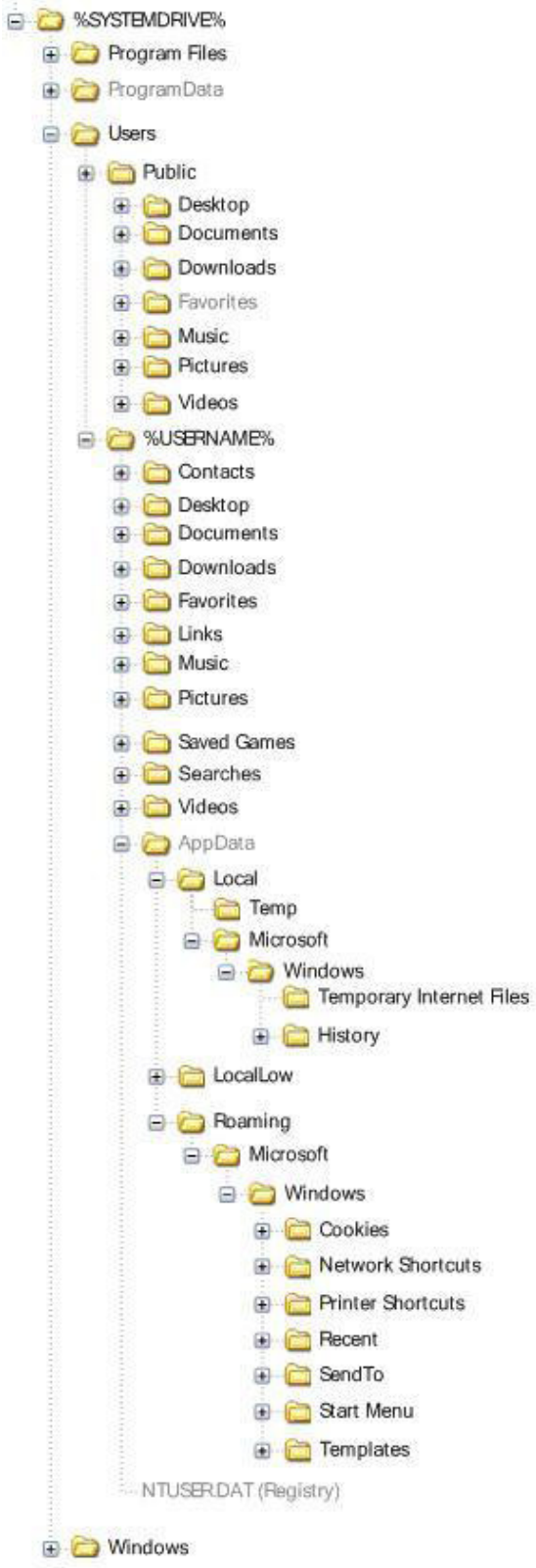
*[4] Namespace Usage Guidelines for the Windows Vista File System.*
*http://www.microsoft.com/downloads/details.aspx?FamilyID=88ad7e7c-4068-48b8-9503-e160a6693bba&DisplayLang=en*

Um sich für den richtigen Speicherort zu entscheiden, ist zunächst in End-User-Managed Data und Application-Managed Data zu unterscheiden.
End-User-Managed Data: (visible)

C:\Users\<*username*>\<ContentType> (z.B.: C:\Users\<*username*>\Documents)
C:\Users\Public\<ContentType> (z.B.: C:\Users\Public\ Documents)

Application-Managed Data: (hidden)

C:\Users\<*username*>\AppData (hidden)
C:\ProgramData (hidden)




**End-User-Managed Data**

- Daten müssen leicht gefunden werden.
- Windows Explorer wird zum Organisieren verwendet.
- Benutzer öffnen und speichern die Daten mit Hilfe der Datei-Öffnen- und -Speichern-Dialoge.
- Die Daten gehören den Benutzern.

Beispiele:
- Microsoft Office Dokumente
- Media-Dateien, Fotos, Musik, Videos

**Per-User Data**
- Private Daten eines best. Users mit Option zur Freigabe einzelner Ordner oder Dateien *C:\Users\<username>\...*

**Shared-User Data**
- Gemeinsam zugängliche Daten aller Users der Maschine oder des Netzwerks *C:\Users\Public*

**Application-Managed Data**

- Applications erstellen und verwalten diese Daten
- Benutzer müssen keine Kenntnis davon haben wann und wo diese erstellt werden und bearbeiten sie nicht direkt
- Die Applikation beinhaltet die wesentlichen Verwaltungsfunktionen zum Erstellen, Löschen und Backup
- Die Applikation stellt Funktionen bereit, um auf die Daten zuzugreifen

Beispiele:
- Outlook® PST Dateien
- Windows Registry (NTUSER.DAT)
- Temp/cache files
- Temporary Internet Files
- Application Settings, INI Files, Datenbanken
- Log files

***C:\Program Files*** Programme und Komponenten

- User sollten keinerlei Daten speichern
- Schreibgeschützt für Standard-Benutzer

***C:\ProgramData*** Anwendungsdaten shared

- Anwendungsdaten aller Benutzer

***C:\Users\<username>\AppData***

- Anwendungsdaten exclusiv pro User

Weitere Details wie C:\Users\<…>\AppData\Roaming, …\Local, …\LocalLow siehe [4].

**Richtlinien zur Bennennung**

..\Vendor Name\Major Product Name (Recommended)
zB.: C:\ProgramData\Microsoft\Office

..\Minor Product Name (Optional)
z.B.: C:\ProgramData\Microsoft\Office\Groove

..\Version + SKU Information (Optional)
z.B.: C:\ProgramData\Microsoft\Office\Groove\2007 Ultimate Edition

Anwendungen sollten keine Dateien oder Verzeichnisse im Root eines Laufwerkes erstellen, können aber die Möglichkeiten zur Folder-Relocation nutzen, die das Betriebssystem bietet. Details siehe [4]

**Umsetzung in der eigenen Access-Anwendung**

Um die Anforderungen umsetzen zu können, ist es erforderlich, sich von hartcodierten Pfadangaben in der Anwendung zu verabschieden.
Vista bietet dazu einen Satz von 'Known Folder' APIs. Neben der schon in Windows 2000/XP verfügbaren SHGetFolderPath Function steht eine neue SHGetKnownFolderPath Function zur Verfügung. Dev Ashish hat eine Access-konforme Version unter

*[5] Retrieving a Special Folder's location*
*http://www.mvps.org/access/api/api0054.htm*

veröffentlicht.

Eine Dokumentation der CSIDL-Konstanten findet man in der MSDN unter
http://msdn2.microsoft.com/en-us/library/ms649274.aspx.

Ebenso stehen die aus XP bekannten, nach wie vor unterstützten, Environment-Variablen zur Ermittlung der Pfade zur Verfügung. In Vista sind zudem drei neue Variablen verfügbar.

• supported in Windows Vista:

• %USERPROFILE% => ..\Users\<*username*>\
• %APPDATA% => ..\Users\<*username*>\AppData\Roaming\
• %TEMP% => ..\Users\<*username*>\AppData\Local\Temp\

• added for Windows Vista:

• %LOCALAPPDATA% => ..\Users\<*username*>\AppData\Local
• %PUBLIC% => ..\Users\Public
• %PROGRAMDATA% => %systemdrive%\ProgramData

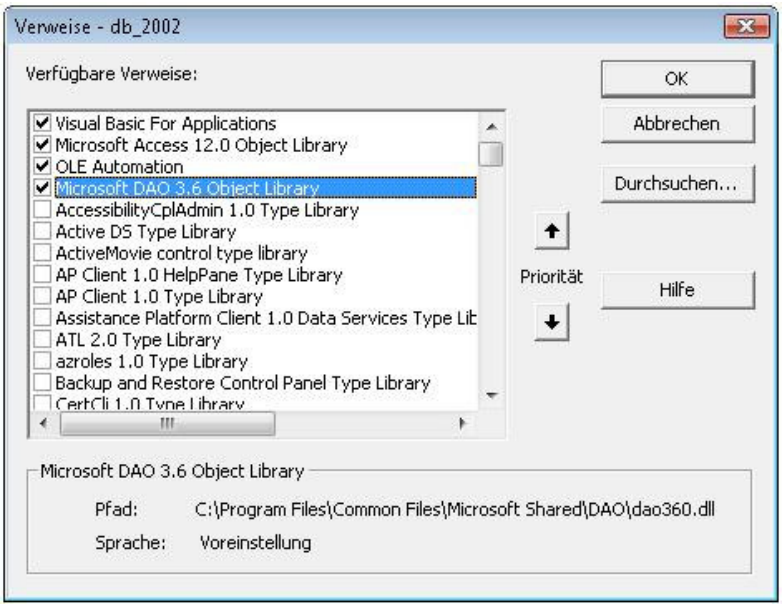
-> Praxisteil: einige Beispiel-Szenarien

## Mehrere Access Versionen auf dem PC - Verweisprobleme

Grundsätzlich empfiehlt es sich, nur eine Office/Access Version auf der Maschine zu installieren, da es laut Microsoft ansonsten Probleme beim automatischen Update gibt.
Eine einfache Möglichkeit bietet ggf. die Verwendung von virtuellen Maschinen wie Microsoft Virtual PC oder VMWare.

Entscheidet man sich trotzdem, verschiedene Versionen zu installieren, gibt es beim Wechsel von Access 2007 zu einer niedrigeren Access-Version (2003/2002/2000) ein Problem mit dem Verweis auf die Microsoft Access XX.0 Object Library. Die Versionen 2000 bis 2007 verwenden den gleichen

Schlüssel unter HKEY_CLASSES_ROOT\TypeLib\{4AFFC9A0-5F99-101B-AF4E}\9.0 um auf die Microsoft Access XX.0 Object Library zu verweisen.
Der Screenshot zeigt den Verweise-Dialog von Access 2002

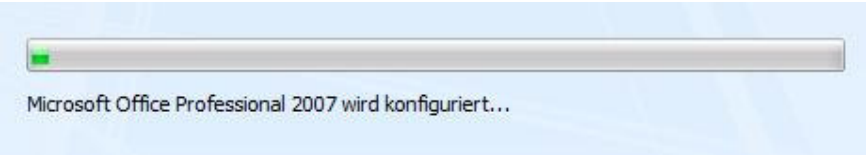



Eine mögliche Lösung wäre die Änderung des zuständigen Registry-Schlüssels per REG-Datei vor dem Start der niedrigeren Access-Version (Pfadangaben sind anzupassen). Ein Start mit Administratorrechten allein reicht nicht aus.

```
Windows Registry Editor Version 5.00

[HKEY_CLASSES_ROOT\TypeLib\{4AFFC9A0-5F99-101B-AF4E-00AA003F0F07}\9.0]
@="Microsoft Access 10.0 Object Library"

[HKEY_CLASSES_ROOT\TypeLib\{4AFFC9A0-5F99-101B-AF4E-00AA003F0F07}\9.0\0]

[HKEY_CLASSES_ROOT\TypeLib\{4AFFC9A0-5F99-101B-AF4E-00AA003F0F07}\9.0\0\win32]
@="C:\\Office10\\Office10\\MSACC.OLB"

[HKEY_CLASSES_ROOT\TypeLib\{4AFFC9A0-5F99-101B-AF4E-00AA003F0F07}\9.0\FLAGS]
@="0"

[HKEY_CLASSES_ROOT\TypeLib\{4AFFC9A0-5F99-101B-AF4E-00AA003F0F07}\9.0\HELPDIR]
@="C:\\Office10\\Office10\\"
```

Wird Access 97 verwendet, empfiehlt es sich, auf den Schlüssel HKEY_CLASSES_ROOT\TypeLib\{4AFFC9A0-5F99-101B-AF4E-00AA003F0F07} sowie Unterordner Schreibzugriffe zuzulassen. Da Access 97 den Unterschlüssel …\8.0 verwendet, wird das Setzen des Verweises korrekt durchgeführt.

Der Wechsel von einer Version < 2007 zu Access 2007 erfordert Geduld, da sich der Installer mit der Reparatur viel Zeit lässt.




Tipp:
Möchte man Access 200x zusammen mit Access 97 betreiben, kann man durch Leeren der MSACCCESS.SRG eine Re-Registrierung von A97 vermeiden.

## Weitere bekannte Probleme in Access-Anwendungen

- Vista blockiert die SendKeys Funktion in A97. Aufrufe von SendKeys müssen aus einer eigenen Anwendung entfernt und durch API Code ersetzt werden.
- Es ist darauf zu achten, dass Anwendungen, die ADO verwenden auf ADO >= Version 2.8 verweisen.
- Es ist darauf zu achten, dass Anwendungen, die XML verwenden auf XML >= Version 3.0 verweisen.
- Die VBA-Funktion DeleteSetting funktioniert nicht mehr ohne die Angabe eines Keys.
- Falls das PlaySound API verwendet wird, sind PCM Files oder MP3-Files in Stereo zu verwenden.

-> Praxisteil: ggf. weitere aktuell in Newsgroups oder Knowledge-Base angesprochene Probleme

## Setup Pakete

- Pfade in den Setup-Paketen sind an die Anforderungen von Vista anzupassen.
- keine hartcodierten Pfade verwenden.
- Zwinge den User zu "Run as Administrator" direkt im Setup-Paket.
- Informiere den User über die Notwendigkeit des Rechtsklicks auf "Run As Administrator".

## Testen der Anwendungs-Kompatibilität

-> Praxisteil: Application Compatibility Toolkits" (ACT)