Thomas-Eßer-Berufskolleg

Beschreibung: C:\Users\Rolf Wippern.RolfWippern-PC\Desktop\TEB Piktogramme\TEB_Berichs-Piktogrammleiste.jpgdes Kreises Euskirchen • Sekundarstufe II

Informationstechnik

BSNW/Windows Klasse

Unterrichtsmaterial: Windows Eingabeaufforderung

1. **Übersicht über die Eingabeaufforderung**

Die Eingabeaufforderung ist ein eigenständiges Softwareprogramm, über das Benutzer direkt mit dem Betriebssystem kommunizieren können. Die nichtgrafische Benutzeroberfläche der Eingabeaufforderung stellt eine Umgebung zur Ausführung zeichenbasierter Anwendungen und Dienstprogramme zur Verfügung. Ähnlich wie der MS-DOS-Befehlsinterpreter **Command.com** führt die Eingabeaufforderung Programme aus und zeigt ihre Ausgabe auf dem Bildschirm unter Verwendung einzelner Zeichen an. Die Windows XP-Eingabeaufforderung verwendet den Befehlsinterpreter **Cmd.exe**, der Anwendungen lädt und den Informationsfluss zwischen Anwendungen steuert, um Benutzereingaben in ein für das Betriebssystem verständliches Format zu übersetzen.

Sie können mithilfe der Eingabeaufforderung Batchdateien (die auch als Skripts bezeichnet werden) erstellen und bearbeiten, um Routineaufgaben zu automatisieren. So können Sie beispielsweise Skripts verwenden, um die Verwaltung von Benutzerkonten oder Sicherungsvorgängen über Nacht zu automatisieren. Darüber hinaus können Sie den Windows Script Host [CScript.exe](MS-ITS:ntcmds.chm::/cscript_overview.htm) verwenden, um komplexere Skripts in der Eingabeaufforderung auszuführen. Durch die Verwendung von Batchdateien können Sie Vorgänge effizienter durchführen als durch die Verwendung der Benutzeroberfläche. Für Batchdateien können alle Befehle verwendet werden, die in der Befehlszeile zur Verfügung stehen. Weitere Informationen zu Batchdateien und zum Skripting finden Sie unter [Verwenden von Batchdateien](MS-ITS:ntcmds.chm::/batch.htm).

Sie können das Eingabeaufforderungsfenster anpassen, um die Lesbarkeit zu erhöhen und die Kontrolle über die Ausführung von Programmen zu steigern. Weitere Informationen zum Anpassen des Eingabeaufforderungsfensters finden Sie unter [So konfigurieren Sie die Eingabeaufforderung](MS-ITS:ntcmds.chm::/CommandPromptOptions.htm).

1. **Verwenden der Befehlssyntax**

Die Syntax ist in der Reihenfolge angegeben, in der Sie einen Befehl und eventuelle nachfolgende Parameter eingeben müssen. Das folgende Beispiel für den Befehl **xcopy** zeigt eine Reihe von Syntaxtextformaten:

**xcopy** *Quelle* [*Ziel*] [**/w**] [**/p**] [**/c**] [**/v**] [**/q**] [**/f**] [**/l**] [**/g**] [**/d**[**:***mm-tt-jjjj*]] [**/u**] [**/i**] [**/s** [**/e**]] [**/t**] [**/k**] [**/r**] [**/h**] [{**/a**|**/m**}] [**/n**] [**/o**] [**/x**] [**/exclude:***Datei1*[**+**[*Datei2*]][**+**[*Datei3*]] [{**/y**|**/-y**}] [**/z**]

Die folgende Tabelle erläutert die Interpretation der verschiedenen Textformate.

1. **Formatting legend**

|  |  |
| --- | --- |
| **Format** | **Meaning** |
| *Italic* | Information that the user must supply |
| **Bold** | Elements that the user must type exactly as shown |
| Ellipsis (...) | Parameter that can be repeated several times in a command line |
| Between brackets ([]) | Optional items |
| Between braces ({}); choices separated by pipe (|). Example: {even|odd} | Set of choices from which the user must choose only one |
| Courier font | Code or program output |

1. **Verwenden mehrerer Befehle und Symbole zur bedingten Verarbeitung**

Sie können mehrere Befehle in einer Befehlszeile oder einem Skript ausführen, wenn Sie Symbole zur bedingten Verarbeitung verwenden. Wenn Sie mehrere Befehle mit Symbolen zur bedingten Verarbeitung ausführen, führen die Befehle auf der rechten Seite des Symbols bestimmte Aktionen in Abhängigkeit von den Ergebnissen auf der linken Seite des Symbols durch. So können Sie beispielsweise einen Befehl nur dann ausführen lassen, wenn der vorangegangene Befehl fehlschlägt. Oder Sie können einen Befehl nur dann ausführen lassen, wenn der vorangegangene Befehl erfolgreich ausgeführt wird.

Sie können die in der folgenden Tabelle aufgeführten Sonderzeichen verwenden, um mehrere Befehle zu übergeben.

|  |  |  |
| --- | --- | --- |
| **Zeichen** | **Syntax** | **Definition** |
| **&** [...] | *Befehl1* **&** *Befehl2* | Verwenden Sie dieses Zeichen, um mehrere Befehle in einer Befehlszeile voneinander zu trennen. **Cmd.exe** führt zunächst den ersten Befehl und anschließend den zweiten Befehl aus. |
| **&&** [...] | *Befehl1* **&&** *Befehl2* | Verwenden Sie dieses Zeichen, um den auf **&&** folgenden Befehl nur dann auszuführen, wenn der Befehl vor dem Symbol erfolgreich ausgeführt wird. **Cmd.exe** führt zunächst den ersten Befehl aus. Der zweite Befehl wird nur dann ausgeführt, wenn der erste Befehl erfolgreich war. |
| **||** [...] | *Befehl1* **||** *Befehl2* | Verwenden Sie dieses Zeichen, um den auf **||** folgenden Befehl nur dann auszuführen, wenn der Befehl vor **||** fehlschlägt. **Cmd.exe** führt zunächst den ersten Befehl aus. Der zweite Befehl wird nur dann ausgeführt, wenn der erste Befehl nicht erfolgreich ausgeführt wird (und ein Fehlercode größer Null zurückgegeben wird). |
| **( )** [...] | **(***Befehl1* **&** *Befehl2***)** | Verwenden Sie diese Zeichen, um mehrere Befehle zu gruppieren oder zu verschachteln. |
| **;** oder **,** | *Befehl1* *Parameter1***;***Parameter2* | Verwenden Sie diese Zeichen, um Befehlsparameter zu trennen. |

ms-its:c:\windows\help\ntshared.chm::/note.gifAnmerkungen

* Das kaufmännische Und (&), das Verkettungszeichen (|) und die runden Klammern ( ) sind Sonderzeichen, denen das Escapezeichen (^) oder Anführungszeichen vorangestellt werden müssen, wenn sie als Argumente übergeben werden sollen.
* Bei erfolgreicher Ausführung einer Operation gibt ein Befehl den Beendigungscode Null (0) oder keinen Beendigungscode zurück. Weitere Informationen zu Beendigungscodes finden Sie unter [Microsoft Windows Resource Kits](ms-its:C:\WINDOWS\Help\ntcmds.chm::/CHM=howto.chm;nofts.chm%20FILE=app_reskit.htm;windows_non_specific.htm).

1. **Verschachteln von Eingabeaufforderungen**

Sie können Eingabeaufforderungen innerhalb von **Cmd.exe** verschachteln, indem Sie an der Eingabeaufforderung eine neue Instanz von **Cmd.exe** öffnen. Standardmäßig erbt jede Instanz von **Cmd.exe** die Umgebung der übergeordneten **Cmd.exe**-Anwendung. Durch die Verschachtelung von **Cmd.exe**-Instanzen können Sie Änderungen an der lokalen Umgebung vornehmen, die sich nicht auf die übergeordnete Anwendung von **Cmd.exe** auswirken. Auf diese Weise können Sie die ursprüngliche Umgebung von **Cmd.exe** beibehalten und nach Beendigung der verschachtelten Eingabeaufforderung zu ihr zurückkehren. Die Änderungen, die Sie in der verschachtelten Eingabeaufforderung vornehmen, werden nicht gespeichert.

Um eine Eingabeaufforderung zu verschachteln, geben Sie an der Eingabeaufforderung Folgendes ein:

**cmd**

Es wird eine ähnliche Meldung wie die folgende angezeigt:

Microsoft (R) Windows XP (TM)  
(C) Copyright 1985-2001 Microsoft Corp.

Um die verschachtelte Eingabeaufforderung zu schließen, geben Sie **exit** ein.

Sie können den Gültigkeitsbereich von Änderungen in einer Instanz von **Cmd.exe** (oder in einem Skript) noch weiter einschränken, indem Sie die Befehle **setlocal** und **endlocal** verwenden. **Setlocal** erstellt einen lokalen Bereich, und **endlocal** beendet den lokalen Bereich. Alle Änderungen, die innerhalb des Bereichs zwischen **setlocal** und **endlocal** vorgenommen wurden, werden verworfen, so dass die ursprüngliche Umgebung unverändert beibehalten wird. Sie können diese beiden Befehle in maximal 32 Ebenen verschachteln. Weitere Informationen zu den Befehlen **setlocal** und **endlocal** finden Sie unter [Setlocal](MS-ITS:ntcmds.chm::/setlocal.htm) und [Endlocal](MS-ITS:ntcmds.chm::/endlocal.htm).

1. **Verwenden von Umgebungsvariablen mit Cmd.exe**

Die Eingabeaufforderungsumgebung **Cmd.exe** wird durch Variablen definiert, die das Verhalten der Eingabeaufforderung und des Betriebssystems bestimmen. Durch die Verwendung von zwei Umgebungsvariablentypen - Systemvariablen und lokalen Variablen - können Sie das Verhalten der Eingabeaufforderungsumgebung oder der gesamten Betriebssystemumgebung definieren. Systemumgebungsvariablen definieren das Verhalten der globalen Betriebssystemumgebung. Lokale Umgebungsvariablen definieren das Verhalten der Umgebung der aktuellen Instanz von **Cmd.exe**.

Systemumgebungsvariablen werden im Betriebssystem voreingestellt und stehen allen Windows XP-Prozessen zur Verfügung. Nur Benutzer mit Administratorrechten können Systemvariablen ändern. Diese Variablen werden am häufigsten in Anmeldeskripts verwendet.

Lokale Umgebungsvariablen stehen nur dann zur Verfügung, wenn der Benutzer, für den sie erstellt wurden, am Computer angemeldet ist. Lokale Variablen, die in der [Struktur](ms-its:C:\WINDOWS\Help\ntcmds.chm::/HELP=glossary.hlp%20TOPIC=gls_registry_hive) **HKEY\_CURRENT\_USER** festgelegt sind, gelten nur für den aktuellen Benutzer, definieren jedoch das Verhalten der globalen Betriebssystemumgebung.

In der folgenden Liste sind die verschiedenen Variablentypen in absteigender Rangfolge aufgeführt.

1. Integrierte Systemvariablen
2. Systemvariablen in der Struktur **HKEY\_LOCAL\_MACHINE**
3. Lokale Variablen in der Struktur **HKEY\_CURRENT\_USER**
4. Alle Umgebungsvariablen und Pfade, die in der Datei **Autoexe.bat** festgelegt sind
5. Alle Umgebungsvariablen und Pfade, die in einem Anmeldeskript (sofern vorhanden) festgelegt sind
6. Variablen, die interaktiv in einem Skript oder einer Batchdatei verwendet werden

In der Eingabeaufforderung erbt jede Instanz von **Cmd.exe** die Umgebung der übergeordneten Anwendung. Sie können daher die Variablen in der neuen **Cmd.exe**-Umgebung ändern, ohne dass hierdurch die Umgebung der übergeordneten Anwendung beeinflusst wird.

In der folgenden Tabelle werden die Systemumgebungsvariablen und die lokalen Umgebungsvariablen für Windows XP aufgeführt.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Typ** | **Beschreibung** |
| %ALLUSERSPROFILE% | lokal | Gibt den Pfad des Profils **All Users** zurück. |
| %APPDATA% | lokal | Gibt den Pfad zurück, in dem Anwendungen Daten standardmäßig speichern. |
| %CD% | lokal | Gibt die Zeichenfolge des aktuellen Verzeichnisses zurück. |
| %CMDCMDLINE% | lokal | Gibt die exakte Befehlszeile zurück, die zum Starten der aktuellen **Cmd.exe** verwendet wird. |
| %CMDEXTVERSION% | System | Gibt die Versionsnummer der aktuellen Befehlserweiterungen zurück. |
| %COMPUTERNAME% | System | Gibt den Namen des Computers zurück. |
| %COMSPEC% | System | Gibt den exakten Pfad zur ausführbaren Datei der Eingabeaufforderung zurück. |
| %DATE% | System | Gibt das aktuelle Datum zurück. Verwendet dasselbe Format wie der Befehl **date /t**. Generiert durch **Cmd.exe**. Weitere Informationen zum Befehl **date** finden Sie unter [Date](MS-ITS:ntcmds.chm::/date.htm). |
| %ERRORLEVEL% | System | Gibt den Fehlercode des zuletzt verwendeten Befehls zurück. Ein Wert ungleich Null weist in der Regel auf einen Fehler hin. |
| %HOMEDRIVE% | System | Gibt den Laufwerkbuchstaben der lokalen Arbeitsstation zurück, die mit dem Basisverzeichnis des Benutzers verbunden ist. Festgelegt basierend auf dem Wert des Basisverzeichnisses. Das Basisverzeichnis des Benutzers wird in **Lokale Benutzer und Gruppen** festgelegt. |
| %HOMEPATH% | System | Gibt den vollständigen Pfad des Basisverzeichnisses des Benutzers zurück. Festgelegt basierend auf dem Wert des Basisverzeichnisses. Das Basisverzeichnis des Benutzers wird in **Lokale Benutzer und Gruppen** festgelegt. |
| %HOMESHARE% | System | Gibt den Netzwerkpfad zum freigegebenen Basisverzeichnis des Benutzers zurück. Festgelegt basierend auf dem Wert des Basisverzeichnisses. Das Basisverzeichnis des Benutzers wird in **Lokale Benutzer und Gruppen** festgelegt. |
| %LOGONSEVER% | lokal | Gibt den Namen des Domänencontrollers zurück, der die aktuelle Anmeldesitzung bestätigt hat. |
| %NUMBER\_OF\_PROCESSORS% | System | Gibt die Anzahl der im Computer installierten Prozessoren an. |
| %OS% | System | Gibt den Namen des Betriebssystems zurück. Windows 2000 zeigt als Betriebssystem Windows\_NT an. |
| %PATH% | System | Gibt den Suchpfad für ausführbare Dateien an. |
| %PATHEXT% | System | Gibt eine Liste der Dateierweiterungen zurück, die vom Betriebssystem als ausführbar angesehen werden. |
| %PROCESSOR\_ARCHITECTURE% | System | Gibt die Chiparchitektur des Prozessors zurück. Werte: [x86](ms-its:C:\WINDOWS\Help\ntcmds.chm::/HELP=glossary.hlp%20TOPIC=x86), IA64. |
| %PROCESSOR\_IDENTFIER% | System | Gibt eine Beschreibung des Prozessors zurück. |
| %PROCESSOR\_LEVEL% | System | Gibt die Modellnummer des im Computer installierten Prozessors zurück. |
| %PROCESSOR\_REVISION% | System | Gibt die Revisionsnummer des Prozessors zurück. |
| %PROMPT% | lokal | Gibt die Einstellungen der Eingabeaufforderung für den aktuellen Interpreter zurück. Generiert durch **Cmd.exe**. |
| %RANDOM% | System | Gibt eine zufällige Dezimalzahl zwischen 0 und 32767 zurück. Generiert durch **Cmd.exe**. |
| %SYSTEMDRIVE% | System | Gibt das Laufwerk zurück, das das Windows XP-Stammverzeichnis enthält (d. h., den Systemstamm). |
| %SYSTEMROOT% | System | Gibt den Pfad des Windows XP-Stammverzeichnisses zurück. |
| %TEMP% und %TMP% | System und Benutzer | Gibt die standardmäßigen temporären Verzeichnisse zurück, die von Anwendungen verwendet werden, die aktuell angemeldeten Benutzern zur Verfügung stehen. Bei einigen Anwendungen muss TEMP, bei anderen TMP verwendet werden. |
| %TIME% | System | Gibt die aktuelle Uhrzeit zurück. Verwendet dasselbe Format wie der Befehl **time /t**. Generiert durch **Cmd.exe**. Weitere Informationen zum Befehl **time** finden Sie unter [Time](MS-ITS:ntcmds.chm::/time.htm). |
| %USERDOMAIN% | lokal | Gibt den Namen der Domäne zurück, die das Konto des Benutzers enthält. |
| %USERNAME% | lokal | Gibt den Namen des Benutzers zurück, der aktuell angemeldet ist. |
| %USERPROFILE% | lokal | Gibt den Pfad des Profils für den aktuellen Benutzer zurück. |
| %WINDIR% | System | Gibt den Pfad des Betriebssystemverzeichnisses zurück. |

1. **Festlegen von Umgebungsvariablen**

Mithilfe des Befehls **set** können Sie Umgebungsvariablen erstellen, ändern, löschen oder anzeigen. Der Befehl **set** ändert Variablen nur in der aktuellen Shellumgebung.

Um eine Variable anzuzeigen, geben Sie an der Eingabeaufforderung Folgendes ein:

**set** *Variablenname*

Um eine Variable hinzuzufügen, geben Sie an der Eingabeaufforderung Folgendes ein:

**set** *Variablenname***=***Wert*

Um eine Variable zu löschen, geben Sie an der Eingabeaufforderung Folgendes ein:

**set** *Variablenname***=**

Sie können die meisten Zeichen als Variablenwerte verwenden, einschließlich Leerzeichen. Wenn Sie die Sonderzeichen <, >, |, &, oder ^ verwenden möchten, müssen Sie ihnen das Escapezeichen (^) oder Anführungszeichen voranstellen. Wenn Sie Anführungszeichen verwenden, werden diese in den Wert eingeschlossen, weil alle Zeichen, die auf das Gleichheitszeichen folgen, als Wert interpretiert werden. Betrachten Sie die folgenden Beispiele:

* Um den Variablenwert **neuer&Name** zu erstellen, geben Sie Folgendes ein:

**set VarName=neuer^&Name**

* Um den Variablenwert **"neuer&Name"** zu erstellen, geben Sie Folgendes ein:

**set VarName="neuer&Name"**

* Wenn Sie an der Eingabeaufforderung **set VarName=neuer&Name** eingeben, wird eine Fehlermeldung ähnlich der folgenden angezeigt:

"Der Befehl 'Name' ist entweder falsch geschrieben oder konnte nicht gefunden werden."

Bei Variablennamen muss keine Groß-/Kleinschreibung beachtet werden. Der Befehl **set** zeigt die Variable jedoch genau so an, wie Sie sie eingegeben haben. Sie können Großbuchstaben und Kleinbuchstaben in Variablennamen kombinieren, um die Lesbarkeit des Codes zu erhöhen (z. B. BenutzerName).

ms-its:c:\windows\help\ntshared.chm::/note.gifAnmerkungen

* Die maximale Variablengröße in einer Umgebung beträgt 8192 Bytes.
* Die maximale Größe aller Umgebungsvariablen insgesamt, einschließlich der Variablennamen und des Gleichheitszeichens, beträgt 65.536 KB.

1. **Ersetzen von Umgebungsvariablenwerten**

Um das Ersetzen von Variablenwerten in der Befehlszeile oder in Skripts zu ermöglichen, schließen Sie den Variablennamen in Prozentzeichen ein (**%***Variablenname***%**). Durch die Verwendung von Prozentzeichen stellen Sie sicher, dass **Cmd.exe** die Variablenwerte referenziert und keinen buchstabengetreuen Vergleich vornimmt. Schließen Sie nach dem Definieren von Variablenwerten für einen Variablennamen den Variablennamen in Prozentzeichen ein. **Cmd.exe** sucht nach allen Instanzen des Variablennamens und ersetzt ihn durch den definierten Variablenwert. Wenn Sie beispielsweise ein Skript erstellen, das verschiedene Werte enthält (z. B. Benutzernamen), und Sie die Umgebungsvariable **USERNAME** für jeden Benutzer mit diesen Werten definieren möchten, können Sie ein Skript erstellen, in dem die Variable **USERNAME** in Prozentzeichen eingeschlossen ist. Bei der Ausführung des Skripts ersetzt **Cmd.exe** **%USERNAME%** durch die Variablenwerte, so dass dieser Vorgang nicht für jeden Benutzer manuell durchgeführt werden muss. Das Ersetzen von Variablen erfolgt nicht rekursiv. **Cmd.exe** überprüft Variablen einmal. Weitere Informationen zum Ersetzen von Variablen finden Sie unter [For](MS-ITS:NTCmds.chm::/for.htm) und [Call](MS-ITS:NTCmds.chm::/call.htm)

Quelle: Windows Hilfe "Windows Eingabeaufforderung"