Prozesse

handle WWW

handle [-a**] [**-u**] [**-c *Handle* **[**-y**]] | [**-s**]] [**-p *Prozessname*| PID **[***Name***]**

Dieses Sysinternals-Tool zeigt offene Handles aller Prozesse (-a) oder nach Benutzernamen des Besitzers (-u) an, um zum Beispiel Performanceengpässen auf die Spur zu kommen. Optional werden nur die Handles eines per Name oder PID angegebenen Prozesses oder eines bestimmten Objekts aufgelistet. Mit -c und als Hexadezimalzahl angegebenem Handle kann dieses geschlossen werden. Dabei ist das Risiko der Destabilisierung des laufenden Betriebssystems zu beachten.

Beispiel

handle -p expl temp
: Listet alle Objekte (Dateien, Registry-Einträge) auf, die gerade von einem Prozess mit *expl* im Namen geöffnet sind und in einem Pfad liegen, der die Zeichenkette *temp* enthält – kurz, alle vom Explorer geöffneten temporären Dateien.

psexec WWW

psexec [*Optionen*] *Befehl*

Dieses leistungsfähige Kommando aus der zu Microsoft gehörenden Toolschmiede Sysinternals startet Kommandozeilenprogramme auf entfernten Systemen, wobei Ein- und Ausgabe auf den lokalen Computer umgeleitet werden. Auf den entfernten Systemen erzeugt psexec über das Netzwerk kurzerhand einen Dienst, der zum Starten des angegebenen *Befehl*s dient und nach dessen Beendigung wieder gelöscht wird.

An dieser Stelle werden nur die wichtigsten Optionen beschrieben. Ein Aufruf des Befehls ohne Parameter liefert Erklärungen zu den umfangreichen Optionen.

Optionen

*****Computer1***[,***Computer2***[,...]] | @***Datei*
Führt den Befehl auf den angegebenen *Computern* oder den in *Datei* angegebenen Computern aus.

-c
Kopiert die angegebene Befehlsdatei auf den entfernten Computer. Falls diese Option nicht angegeben wird, muss der verwendete Befehl im Pfad des entfernten Systems liegen. Alternativ kann der volle Pfad zum Befehl aus Sicht des entfernten Systems angegeben werden.

-w *Arbeitsverzeichnis*
Setzt das *Arbeitsverzeichnis* für den Befehl relativ zum entfernten System.

[-u *Benutzer* **[-p** *Kennwort***]]**
Verwendet nicht den aktuellen, sondern den angegebenen Benutzer. Wird kein *Kennwort* angegeben, wird es an der Eingabeaufforderung abgefragt.

PoSh: PowerShell ermöglicht den Remotezugriff auf andere Rechner. Das gilt zum einen für Cmdlets, die sich über Computernamen auf andere Systeme beziehen, und zum anderen für komplette PowerShell-Sitzungen.

Ist der Zielrechner korrekt konfiguriert und Mitglied derselben Domäne und hat der Benutzer das Recht zur lokalen Anmeldung auf dem Zielrechner, können Sie mit dem folgenden Befehl eine Konsolensitzung direkt auf dem Remoterechner eröffnen:

```
new-pssession -computername Zielrechner -credential Domänenname\
Benutzername
```

Sie beenden die Sitzung mit `exit-pssession`.

runas

runas [*Optionen*] /user:*Benutzername Befehl*

Führt den angegebenen Befehl im Kontext des angegebenen Benutzers aus.

runas [[*/noprofile* | */profile*] [*/env*] [*/savecred* | */netonly*]]
/user:*Benutzername* ***Programm***

runas [[*/noprofile* | */profile*] [*/env*] [*/savecred*]]
/smartcard **[*/user:****Benutzername*] ***Programm***

runas ***/trustlevel:****Vertrauensstufe* ***Programm***

Optionen

/env
: Verwendet die aktuelle Umgebung statt der des angegebenen Benutzers. Dadurch enthält z.B. die Umgebungsvariable `%UserProfile%` den Pfad nicht zum Profil des angegebenen, sondern des aufrufenden Benutzers.

/netonly
: Der angegebene Benutzer muss nur über das Recht zur Anmeldung über das Netzwerk verfügen, nicht über das Recht zur lokalen Anmeldung.

/noprofile
: Lädt das Benutzerprofil des angegebenen Benutzers nicht, was bei einigen Anwendungen zu Fehlern führen kann. Ohne Angabe dieser Option wird das Profil geladen.

/savecred
: Speichert das Kennwort bei der Eingabe bzw. fragt nicht nach dem Kennwort des angegebenen Benutzers, falls es zuvor gespeichert wurde. Diese Option ist unter Sicherheitsaspekten bedenklich.

/smartcard
: Verwendet auf einer Smartcard gespeicherte Anmeldeinformationen.

/showtrustlevels
: Zeigt die definierten Vertrauensstufen, die für die Option `/trustLevel` verwendbar sind.

/trustLevel:*Stufe*
: Führt einen Befehl in der angegebenen Vertrauensstufe aus, um beispielsweise als Administrator Programme mit eingeschränkten Rechten zu starten.

Falls `/netonly` nicht angegeben wird, muss der angegebene Benutzer über die Berechtigung zur lokalen Anmeldung verfügen.

Das Kennwort des angegebenen Benutzers wird an der Eingabeaufforderung abgefragt. Es ist keine Möglichkeit bekannt, es automatisiert zu übergeben (mit Ausnahme der Verwendung von `/savecred`), was den Nutzen von `runas` in Skripten stark einschränkt.

Sollten einzelne Befehle nicht wie gewünscht arbeiten, kann es hilfreich sein, zunächst eine Instanz von *cmd.exe* im gewünschten Kontext aufzurufen und von hier aus das gewünschte Programm zu starten.

PoSh: PowerShell ermöglicht die Weitergabe oder Abfrage von Anmeldeinformationen mit dem Parameter `-credential 'Domäne\Benutzername'`. Dieser fragt entweder per Pop-up-Dialog nach dem Kennwort des angegebenen Benutzerkontos, oder es wird von vornherein ein Credential-Objekt hinterlegt und dem Befehl mitgegeben. Die Hochkommata sind dringend empfohlen, damit die Sonderzeichen im Kennwort nicht zu Irritationen der PowerShell-Befehle führen.

```
$benutzer = 'Domäne\Admin'
$kennwort = 'Start@3$'
$securePassword = convertto-securestring $kennwort -asplaintext
-force
$credential = new-object System.Management.Automation.PSCredential
$benutzer, $securePassword
Start-Process notepad.exe -credential $credential
```

Warnung: Sie können das Credential-Objekt ebenfalls in Dateiform abspeichern, aber auch wenn der Inhalt von `$securePassword` nicht länger das Kennwort im Klartext enthält, kann es in falsche Hände gelangen und missbraucht werden.

START

START [*"Fenstertitel"*] [***Optionen***] *Befehl Parameter*

Führt einen Befehl aus. Auf Beendigung der Ausführung kann gewartet werden, Prozesspriorität und Prozessoraffinität können festgelegt werden.

Optionen

/d *Verzeichnispfad*
Setzt das Arbeitsverzeichnis für den Befehl.

/b
Startet den Befehl als Hintergrundprozess, erstellt also kein neues Fenster.

/i
Übergibt die aktuelle Umgebung an den gestarteten Prozess.

/low* | */belownormal* | */normal* | */abovenormal* | */high* | */realtime
Legt die Priorität des neuen Prozesses fest.

/affinity *Prozessor*
Setzt die Nummer des Prozessors, auf dem der neue Prozess ausgeführt werden soll (als Hexadezimalzahl).

/node
Legt den Knoten der bevorzugten NUMA (*Non-Uniform Memory Architecture*) als ganzzahligen Dezimalwert fest. Bei Kombination mit der Affinitätsmaske des Prozessors wird diese abweichend interpretiert.

/wait
Startet den Befehl und wartet auf seine Beendigung.

/min* | */max
Startet den Befehl in einem minimierten oder maximierten Fenster.

Die Optionen `/separate` zum Starten in einem separaten und `/shared` zum Starten in einem gemeinsam genutzten Speicherbereich werden auf 64-Bit-Versionen von Windows nicht unterstützt, da diese Optionen ausschließlich der Steuerung des Verhaltens von 16-Bit-Anwendungen dienen.

tasklist

tasklist [/s *System* **[/u** *Benutzer***name [/p [***Kennwort***]]]] [/m [***Modul***]** | **/svc** | **/v] [/fi** *Filter***] [/fo** *Format***] [/nh]**

Zeigt ähnlich wie der Task-Manager laufende Prozesse und deren Speichernutzung auf dem angegebenen oder lokalen System an (*Benutzer* und *Kennwort* können zur Verbindung angegeben werden). Zusätzlich können mit `/m` die von den einzelnen Prozessen gela-

denen DLLs aufgelistet bzw. nach Prozessen gefiltert werden, die ein bestimmtes *Modul* geladen haben. In einem Prozess enthaltene Dienste können angezeigt (`/svc`) oder detailliertere Informationen (`/v`) ausgegeben werden. Mit der Option `/fo` wird das Ausgabeformat festgelegt: tabellarisch (`table`), Listenform (`list`) oder CSV-Format (`csv`). `/nh` unterdrückt die Ausgabe von Spaltenüberschriften.

Zur Definition von Filtern sei auf die Hilfe des Befehls verwiesen.

PoSh: Mit dem PowerShell-Cmdlet `get-process` erhalten Sie eine leistungsstarke Methode zum Auflisten und Filtern laufender Prozesse. So listet der folgende Befehl die mit *c* beginnenden Prozessnamen mit Prozess-ID und Anzahl der durch diesen Prozess verwendeten Handles auf:

```
get-process c* | format-table Name, ID, handles -autosize
```

taskkill

taskkill [/fi *Filter*] **[/pid** *Prozess-ID* | **/im** *EXE-Datei*] **[/s** *System* **[/u** *Benutzer* **[/p** [*Passwort*]]]] **[/f] [/t]**

Beendet einen oder mehrere Prozesse lokal oder auf einem Remotesystem, zu dem mit dem angegebenen Benutzernamen eine Verbindung hergestellt werden kann. Die zu beendenden Tasks werden durch ihre PID (*Prozess-ID*) oder den Namen der *EXE-Datei* ausgewählt, alternativ oder zusätzlich kann ein *Filter* angegeben werden, dessen Syntax der des `tasklist`-Befehls entspricht. `/f` erzwingt das Beenden. Mit `/t` wird der ganze Prozessbaum beendet, also auch die vom angegebenen Prozess gestarteten Kindprozesse.

Zur Definition von Filtern sei auf die Hilfe des Befehls verwiesen.

PoSh: Prozesse lassen sich mit der Windows PowerShell auf verschiedene Weise beenden. Stellvertretend wird an dieser Stelle das Cmdlet `stop-process` aufgeführt.

```
stop-process -name iexplore
```

Beendet alle laufenden Internet-Explorer-Instanzen. Anstelle des Prozessnamens können Sie auch die Prozess-ID angeben. Mit `-force` als zusätzlichem Parameter wird das Schließen von Prozessen erzwungen.

unlodctr

unlodctr Argumente

Löscht Namen und Beschreibungen für erweiterbare Leistungsindikatoren.

Argumente

Treiber
: Gibt den Namen des Gerätetreibers an, dessen Leistungsindikatoren-Namen und -Erklärungen aus der Registrierung entfernt werden sollen.

/m:*Manifest*
: Gibt den Namen der Manifestdatei an, deren enthaltene Leistungsindikatoren-Definitionen aus der Registrierung entfernt werden sollen.

/g:*{Anbieter-GUID}* | **/p:***Anbietername*
: Durch die Anbieter-GUID oder den Anbieternamen wird der entladene Leistungsindikatoren-Anbieter angegeben.

Dienste

net continue

net continue *Dienst*

Setzt die Ausführung eines angehaltenen Diensts fort.

net pause

net pause *Dienst*

Hält einen laufenden Dienst vorübergehend an.

net start

net start [*Dienst***]**

Startet einen Dienst. Falls der angegebene Dienstname Leerzeichen enthält, muss er in Anführungszeichen gesetzt werden. Ohne Para-

meter gibt dieser Befehl die aktiven Dienste mit ihren internen Namen aus.

net stop

net stop *Dienst*

Beendet den angegebenen Dienst.

sc

sc [*Server***]** *Befehl Dienst [Optionen]*

Leistungsstarker Befehl zur Verwaltung von Diensten. Beachten Sie bei der Eingabe des Befehls, dass nach Gleichheitszeichen zur Angabe von Optionen ein Leerzeichen vor dem Wert erforderlich sein kann.

Befehle

query | queryex [Dienstname] [type= {*service* | *driver* | *all*}] [state= [*all* | *inactive*] [bufsize= *Puffergröße***] [ri=***Index***] [group=** *Dienstgruppe***]**

Zeigt ohne Angabe von Dienstnamen den Status (mit queryex den erweiterten Status) aller aktiven Dienste und Treiber an. Durch Angabe eines Typbezeichners (service, driver, all, Standardwert ist service) mit type kann die Liste der angezeigten Dienste gefiltert werden. Die Angabe des Kriteriums state (Status) ermöglicht das Filtern nach aktiven (ist Standard ohne Angabe eines Parameters), inaktiven oder allen Zuständen. bufsize ändert die Größe des Auflistungspuffers (Standard sind 4.096 Byte), ri legt die Indexnummer zum Fortsetzen fest, bei der die Auflistung beginnen soll, und mit group ist die Filterung nach Dienstgruppen möglich.

start *Dienstname* [*Argumente*]

Startet den angegebenen Dienst.

stop *Dienstname* [*Grund***] [***Kommentar***]**

Beendet einen laufenden Dienst und hält optional einen angegebenen Grund sowie einen Kommentar im Ereignislog fest. Der Grund wird als Aneinanderreihung numerischer Werte aus

einer in der Hilfe verfügbaren Liste angegeben, Kennung, Hauptgrund und weiterer Grund werden per Doppelpunkt miteinander verbunden. So bedeutet *1:4:6* Ungeplant, Software, Aufgehängt. Der Kommentar darf maximal 127 Zeichen lang sein.

pause | continue *Dienstname*

Der pause-Befehl hält einen Dienst an, continue setzt ihn wieder fort.

config *Dienstname* [type= {*own* | *share* | *interact* | *kernel* | *filesys* | *rec* | *adapt* | *userown* | *usershare*}] [start= {*boot* | *system* | *auto* | *demand* | *disabled* | *delayed-auto*}] [error= {*normal* | *severe* | *critical* | *ignore*}] [binpath= *Binärpfadname zur EXE-Datei*] [group= *Dienstgruppe*] [tag= {*yes* | *no*}] [depend= *Abhängigkeiten* (Trennung durch Schrägstrich) [obj= {*Kontenname* | *Objektname*}] [displayname= *Anzeigename*] [password= *Kennwort*]

Mit der Operation config ändert sc Einträge zu einem Dienst in der Registrierung und der Dienstdatenbank. Die Optionen entsprechen den wesentlichen Angaben zu Diensten in Windows, beispielsweise können der Typ des Diensts (type), sein Startverhalten (start), die Relevanz eines Fehlers (error), der Pfad der ausführbaren Datei (binpath) und der Anzeigename (displayname) festgelegt werden.

description [*Dienstname*] [*Beschreibung*]

Ändert die Beschreibung eines Diensts.

failure [*Dienstname*] [reset= (*Zeitraum ohne Fehler in Sekunden, nach dem der Fehlerzähler auf 0 gesetzt wird*)] [reboot= (*vor dem Neustart übertragene Nachricht*)] [command= (*bei Fehler auszuführende Befehlszeile*)] [actions= {*run* | *restart* | *reboot*}/(*Verzögerungszeit in Millisekunden*)]

Legt fest, was bei einem Fehler des Diensts passiert: den Zeitraum, nach dem der Fehlerzähler auf 0 gesetzt wird (reset), im Fehlerfall auszuführende Befehle (command) und im Zusammenhang mit der reset-Option, welche Aktionen ausgeführt werden sollen (actions). Bei letzterer Operation werden sowohl die einem Fehler folgende Aktion und Verzögerungszeit (in Millisekunden) bis zu ihrer Ausführung als auch Folgeaktionen durch Schrägstrich getrennt, z. B. run/5000/reboot/1000.

failureflag [*Dienstname*] [*Flag*]
: Ändert das Flag, das das Verhalten eines Diensts bei einem auftretenden Fehler bestimmt. Bei der Standardeinstellung `0` werden konfigurierte Vorgänge zur Fehlerbehandlung vom Dienststeuerungsmanager nur aktiviert, wenn der Dienst nicht den Status *Stopped* aufweist. Wird die Einstellung `1` gesetzt, erfolgt die Aktivierung konfigurierter Vorgänge nur, wenn zusätzlich zum Status *Stopped* der Win32-Beendigungscode nicht auf `1` gesetzt ist.

sidtype [*Dienstname*] [type= {*none* | *unrestricted* | *restricted*}
: Verändert den SID-Typ des angegebenen Diensts und fügt die SID dem Dienstprozesstoken hinzu. Die Einstellung `unrestricted` (unbeschränkt) gilt nur für Dienste, die im Win32-Benutzermodus laufen. Mit der Einstellung `restricted` (eingeschränkt), die ebenfalls nur für Dienste im Win32-Benutzermodus gilt, wird die SID des Diensts in die Liste der eingeschränkten SIDs im Prozesstoken eingetragen. Die Angabe `none` fügt den Dienst nicht dem Prozesstoken für Dienste hinzu. Diese Einstellungen treten erst in Kraft, wenn der betreffende Dienst neu gestartet wird.

privs [*Dienstname*] [*Berechtigungen*]
: Ändert die erforderlichen *Berechtigungen* für einen Dienst. Diese können abweichend von den Berechtigungen für das für den Start des Diensts verwendete Konto konfiguriert werden. Mehrere Berechtigungen werden durch Schrägstrich voneinander getrennt angegeben.

managedaccount [*Dienstname*] [type= {*true* | *false*}]
: Konfiguriert, ob das für den Dienststart verwendete Konto ein verwaltetes Kennwort verwendet. In der Einstellung `true` wird das Kennwort beim Start des Diensts von Netlogon angefordert, bei `false` wird das konfigurierte Kennwort verwendet.

qc [*Dienstname*] *Puffergröße*
: Fragt die Konfiguration eines Diensts ab.

qdescription [*Dienstname*]
: Fragt die Beschreibung eines Diensts ab.

qfailure [*Dienstname*] *Puffergröße*
Fragt die Aktionen ab, die bei Fehlschlagen eines Diensts erfolgen.

qfailureflag [*Dienstname*]
Fragt das Fehleraktionskennzeichen eines Diensts ab.

qsidtype [*Dienstname*]
Fragt den SID-Typ eines Diensts ab.

qprivs [*Dienstname*]
Fragt die erforderlichen Rechte eines Diensts ab.

qprotection [*Dienstname*]
Fragt die Schutzebene eines Diensts ab.

qtriggerinfo [*Dienstname*]
Fragt die Auslöseparameter eines Diensts ab.

qrunlevel [*Dienstname*]
Fragt die niedrigste Ausführungsebene für einen Dienst ab.

qmanagedaccount [*Dienstname*]
Fragt ab, ob der Dienst ein Konto verwendet, das vom LSA verwaltet wird.

qpreferrednode [*Dienstname*]
Fragt den bevorzugten NUMA-Knoten für einen Win32-Dienst ab.

quserservice [*Dienstvorlagenname*]
Fragt die von der angegebenen Benutzerdienstvorlage erstellte Benutzerdienstinstanz des aktuellen Benutzers in der aktuellen Sitzung ab.

delete [*Dienstname*]
Löscht einen Diensteintrag aus der Registrierung. Läuft der Dienst oder wird er durch einen anderen Prozess offen gehalten, wird er zum Löschen markiert.

create [*Dienstname*] [binpath=] [type= {*own* | *share* | *interact* | *kernel* | *filesys* | *rec*}] [start= {*boot* | *system* | *auto* | *demand* | *disabled* | *delayed-auto*}] [error= {*normal* | *severe* | *critical* | *ignore*}] [group= *Dienstegruppe*] **[tag= {*yes* | *no*}] [depend=** *Abhängigkeit1/Abhängigkeit2/...*] **[obj= {***Kontoname* | *Objektname*}] [displayname=** *Anzeigename*] **[password=** *Kennwort*]
Erstellt einen neuen Dienst durch Anlegen der zugehörigen Einträge in der Registrierung und der Dienstdatenbank. Neben dem *Namen* des Diensts ist der *Pfad* zur zugehörigen ausführ-

baren Datei in der Option `binpath` anzugeben. Mit `type` legen Sie die Art fest, auf die der Dienst mit dem Betriebssystem zusammenwirkt, `start` bestimmt durch Einstellen des Starttyps, ob und wann der Dienst gestartet wird, und `error` legt die Bedeutung eines Fehlers für das Betriebssystem fest.

`control [`*`Dienstname`*`]` *`Wert`*
Sendet als *Wert* einen dienstspezifischen Steuerungscode oder einen der Befehle `paramchange`, `netbindadd`, `netbindremove`, `netbindenable`, `netbinddisable` an einen Dienst.

`sdshow [`*`Dienstname`*`]` *`Rechteanzeigen`*
Zeigt die Sicherheitsbeschreibung eines Diensts im SDDL-Format an.

`sdset [`*`Dienstname`*`]` *`Sicherheitsbeschreibung`*
Setzt die Sicherheitsbeschreibung eines Diensts im SDDL-Format.

`showsid [`*`Dienstname`*`]`
Zeigt die SID an, die einem beliebigen Dienstnamen zugeordnet ist.

`triggerinfo [`*`Dienstname`*`]` *`Option1`* **`[`***`Option2 ...`***`]`**
Dient der Änderung der Auslöseparameter eines Diensts, die bestimmen, unter welchen Umständen ein Dienst gestartet oder beendet wird, und ermöglicht die Löschung dieser Parameter.

`preferrednode [`*`Dienstname`*`]` *`Knotennummer`*
Setzt den bevorzugten NUMA-Knoten eines Win32-Diensts. Es muss sich um einen eigenen Verarbeitungsdienst handeln, und die angegebene Knotennummer muss gültig sein. Zum Löschen eines zugewiesenen NUMA-Knoten verwenden Sie den Wert `-1`.

`runlevel [`*`Dienstname`*`]` *`Nummer der Ausführungsebene`*
Ändert die niedrigste Ausführungsebene für einen Dienst. Dieser muss eine der Startarten *Automatisch*, *Manuell* oder *Deaktiviert* nutzen oder ein Treiberdienst ohne PnP sein. Die Ausführungsebene eines Diensts kann nicht niedriger sein als die Ausführungsebene eines Diensts, von dem er abhängt. Wird `0` als Ausführungsebene angegeben, wird die eingestellte Ausführungsebene gelöscht.

`getdisplayname [`*`Dienstname`*`]` **`[`***`Puffergröße`***`]`**
Ermittelt den Anzeigenamen eines Diensts.

getkeyname [*Anzeigename des Diensts*] [*Puffergröße*]
Ermittelt aus dem gegebenen Anzeigenamen des Diensts den Dienstnamen. Schließen Sie Anzeigenamen mit Leerzeichen in Anführungszeichen ein.

enumdepend [*Dienstname*] [*Puffergröße*]
Zeigt von diesem Dienst abhängige Dienste an. Die Puffergröße muss gegebenenfalls erhöht werden, um alle Informationen anzuzeigen.

boot {*ok* | *bad*}
Legt fest, ob die beim aktuellen Systemstart des lokalen Computers verwendete Konfiguration als letzte als funktionierend bekannte Konfiguration (»last known good configuration«) gespeichert werden soll.

lock
Sperrt die Dienstdatenbank des lokalen Systems.

querylock
Fragt den Sperrstatus der Dienstdatenbank ab.

PoSh: Für die allgemeine Diensteverwaltung auf lokalen und entfernten Rechnern existieren in Windows PowerShell folgende Cmdlets:

- `get-service` zum Abrufen der Informationen zu installierten Diensten.
- `start-service`, `restart-service` und `stop-service` zum Starten, Neustarten bzw. Beenden von Diensten.
- `suspend-service` zum Pausieren laufender Dienste und `resume-service` zum Fortsetzen pausierter Dienste.
- `new-service` zum Erstellen eines neuen Diensts.
- `set-service` zum Ändern der Dienstekonfiguration.
- Weitere Cmdlets beziehen sich auf spezielle, anwendungsbezogene Dienste.

Um einen Dienst zu löschen, können Sie die folgende Syntax verwenden:

```
(gwmi win32_service -filter "name='Dienstname'").delete()
```