

## 4.2 Dienststatus und Startart prüfen

Nachdem der betroffene Dienst eindeutig bestimmt wurde, wird geprüft, ob er:

- im Betriebssystem registriert beziehungsweise geladen ist,
- aktuell ausgeführt wird,
- beendet oder fehlgeschlagen ist,
- beim Systemstart oder bei Bedarf gestartet werden soll,
- deaktiviert wurde,
- unter dem vorgesehenen Benutzerkonto läuft,
- nach einem Fehler automatisch neu gestartet wird.

### “ Grundsatz:

Zuerst den vorhandenen Zustand vollständig erfassen. Einen Dienst erst danach und nur bei betrieblicher Freigabe starten, stoppen oder neu starten.

### Ziele dieser Seite

Nach dieser Seite sollst du:

- Dienstname und Anzeigename unterscheiden können,
- den aktuellen Dienststatus ermitteln können,
- die konfigurierte Startart überprüfen können,
- Startart und aktuellen Laufzustand getrennt bewerten können,
- Dienstkonto, Prozess-ID und Exitcode ermitteln können,
- Windows-Dienste, systemd-Units und launchd-Jobs unterscheiden können,
- deaktivierte oder maskierte Dienste erkennen können,
- einen wiederholt abstürzenden Dienst identifizieren können,
- Statusinformationen vor einer Änderung dokumentieren können.

### Sicherheitskennzeichnungen

| Kennzeichnung | Bedeutung                                                         |
|---------------|-------------------------------------------------------------------|
| [RO]          | Rein lesende Abfrage                                              |
| [TEST]        | Aktiver Test ohne beabsichtigte dauerhafte Konfigurationsänderung |
| [PRIV]        | Erfordert möglicherweise erhöhte Rechte                           |
| [FILE]        | Greift lesend oder schreibend auf eine Datei zu                   |
| [SENS]        | Ausgabe kann sensible Informationen enthalten                     |

| Kennzeichnung | Bedeutung                                                    |
|---------------|--------------------------------------------------------------|
| [CHANGE]      | Verändert den Dienstzustand oder die Konfiguration           |
| [DISRUPT]     | Kann laufende Verbindungen oder Arbeitsvorgänge unterbrechen |

## 1. Startart und Dienststatus nicht verwechseln

Die **Startart** beschreibt, wann beziehungsweise wodurch ein Dienst gestartet werden soll. Der **Dienststatus** beschreibt seinen aktuellen Zustand.

| Startart beziehungsweise Aktivierung     | Aktueller Zustand                | Mögliche Bewertung                                                 |
|------------------------------------------|----------------------------------|--------------------------------------------------------------------|
| Automatisch aktiviert                    | Läuft                            | Normaler Zustand                                                   |
| Automatisch aktiviert                    | Beendet                          | Startfehler, Absturz oder bewusster Stopp möglich                  |
| Manuell beziehungsweise bedarfsgesteuert | Beendet                          | Kann ein normaler Zustand sein                                     |
| Manuell beziehungsweise bedarfsgesteuert | Läuft                            | Dienst wurde angefordert oder manuell gestartet                    |
| Deaktiviert                              | Beendet                          | Erwarteter Zustand                                                 |
| Deaktiviert                              | Läuft                            | Zustandsänderung möglicherweise noch nicht wirksam oder Sonderfall |
| Aktiviert                                | Wiederholt startend und stoppend | Absturzschleife oder fehlerhafte Startbedingung möglich            |

“ Ein beendeter Dienst ist nicht automatisch fehlerhaft. Bedarfsgesteuerte Dienste laufen häufig nur, wenn sie tatsächlich benötigt werden.

## 2. Dienstverwaltung der Betriebssysteme vergleichen

| Betriebssystem    | Dienstverwaltung        | Konfigurationseinheit   | Typische Statusabfrage        |
|-------------------|-------------------------|-------------------------|-------------------------------|
| Windows           | Service Control Manager | Windows-Dienst          | Get-Service oder sc.exe query |
| Linux mit systemd | systemd                 | Unit, meistens .service | systemctl status              |

| Betriebssystem | Dienstverwaltung | Konfigurationseinheit          | Typische Statusabfrage       |
|----------------|------------------|--------------------------------|------------------------------|
| macOS          | launchd          | Daemon, Agent oder XPC-Service | <code>launchctl print</code> |

### Wichtige Unterschiede:

- Windows verwendet unter anderem die Startarten automatisch, manuell und deaktiviert.
- systemd unterscheidet den aktuellen Laufzustand von der Aktivierung für den Systemstart.
- Eine systemd-Unit kann zusätzlich `masked` sein. Dann kann sie nicht normal gestartet werden.
- macOS verwendet häufig bedarfsgesteuerte Dienste.
- Ein launchd-Job muss deshalb nicht dauerhaft als Prozess laufen.
- launchd unterscheidet verschiedene Ausführungsbereiche, sogenannte Domains.
- Ein fehlender Prozess bedeutet bei einem bedarfsgesteuerten launchd-Dienst nicht automatisch einen Fehler.

## 3. Dienststatus unter Windows prüfen

`<DIENSTNAME>` muss durch den internen Dienstnamen ersetzt werden.

### Status eines bestimmten Dienstes:

```
[RO] Get-Service -Name "<DIENSTNAME>"
```

### Ausgewählte Statusinformationen anzeigen:

```
[RO] Get-Service -Name "<DIENSTNAME>" |  
Select-Object Name, DisplayName, Status, ServiceType
```

### Mit einem Teil des Dienstnamens suchen:

```
[RO] Get-Service -Name "*<SUCHBEGRIFF>*" -ErrorAction SilentlyContinue
```

### Mit einem Teil des Anzeigenamens suchen:

```
[RO] Get-Service -DisplayName "*<SUCHBEGRIFF>*" -ErrorAction SilentlyContinue
```

Alle beendeten Dienste anzeigen:

```
[RO] Get-Service |  
  Where-Object Status -eq "Stopped" |  
  Sort-Object DisplayName
```

Alle laufenden Dienste anzeigen:

```
[RO] Get-Service |  
  Where-Object Status -eq "Running" |  
  Sort-Object DisplayName
```

Dienststatus mit `sc.exe` abfragen:

```
[RO] sc.exe query "<DIENSTNAME>"
```

Erweiterte Statusabfrage einschließlich Prozess-ID:

```
[RO] sc.exe queryex "<DIENSTNAME>"
```

Typische Zustände sind:

| Zustand                                                              | Bedeutung                    |
|----------------------------------------------------------------------|------------------------------|
| <code>Running</code> beziehungsweise <code>RUNNING</code>            | Dienst wird ausgeführt       |
| <code>Stopped</code> beziehungsweise <code>STOPPED</code>            | Dienst wird nicht ausgeführt |
| <code>StartPending</code> beziehungsweise <code>START_PENDING</code> | Startvorgang läuft           |
| <code>StopPending</code> beziehungsweise <code>STOP_PENDING</code>   | Beendigung läuft             |
| <code>PausePending</code>                                            | Pausierung wird vorbereitet  |
| <code>Paused</code>                                                  | Dienst ist pausiert          |
| <code>ContinuePending</code>                                         | Fortsetzung wird vorbereitet |

“ Bleibt ein Dienst ungewöhnlich lange in `StartPending` oder `StopPending`, müssen Prozess, Abhängigkeiten und Ereignisprotokolle untersucht werden. Ein erzwungener Abbruch ist nicht der erste Diagnoseschritt.

## 4. Startart und Dienstkonto unter Windows prüfen

`Get-Service` zeigt den Laufzustand, aber nicht alle Konfigurationsinformationen. Dafür kann die CIM-Klasse `Win32_Service` verwendet werden.

### Vollständige Basiskonfiguration eines Dienstes:

```
[RO][SENS] Get-CimInstance Win32_Service -Filter "Name='<DIENSTNAME>'" |  
    Select-Object Name,  
        DisplayName,  
        State,  
        Status,  
        StartMode,  
        StartName,  
        ProcessId,  
        ExitCode,  
        ServiceSpecificExitCode,  
        PathName
```

### Nur Startart und Dienstkonto anzeigen:

```
[RO] Get-CimInstance Win32_Service -Filter "Name='<DIENSTNAME>'" |  
    Select-Object Name, State, StartMode, StartName
```

### Konfiguration mit `sc.exe` prüfen:

```
[RO][SENS] sc.exe qc "<DIENSTNAME>"
```

Typische Werte für `StartMode`:

| Wert     | Bedeutung                                                           |
|----------|---------------------------------------------------------------------|
| Auto     | Automatischer Start                                                 |
| Manual   | Manueller beziehungsweise bedarfsgesteuerter Start                  |
| Disabled | Dienst ist deaktiviert                                              |
| Boot     | Start durch den Systemlader, normalerweise bei Treibern             |
| System   | Start während der Kernelinitialisierung, normalerweise bei Treibern |

Typische Angaben von `sc.exe qc`:

| Angabe                          | Bedeutung                                           |
|---------------------------------|-----------------------------------------------------|
| <code>SERVICE_NAME</code>       | Interner Dienstname                                 |
| <code>TYPE</code>               | Art des Dienstes oder Treibers                      |
| <code>START_TYPE</code>         | Konfigurierte Startart                              |
| <code>ERROR_CONTROL</code>      | Verhalten bei Startfehlern während des Systemstarts |
| <code>BINARY_PATH_NAME</code>   | Programmdatei und Startparameter                    |
| <code>DEPENDENCIES</code>       | Konfigurierte Abhängigkeiten                        |
| <code>SERVICE_START_NAME</code> | Verwendetes Dienstkonto                             |

“ `[SENS]`: Programmpfade und Startparameter können interne Verzeichnisstrukturen, Servernamen oder andere schützenswerte Konfigurationsangaben enthalten.

5. Verzögerten automatischen Start unter Windows prüfen

Die Startart `Auto` allein zeigt nicht immer, ob der automatische Start verzögert erfolgt.

Dienstkonfiguration über CIM prüfen:

```
[RO] Get-CimInstance Win32_Service -Filter "Name='<DIENSTNAME>'" |  
Select-Object Name, State, StartMode, DelayedAutoStart
```

Interpretation:

| <code>StartMode</code> | <code>DelayedAutoStart</code> | Bedeutung                               |
|------------------------|-------------------------------|-----------------------------------------|
| <code>Auto</code>      | <code>False</code>            | Normaler automatischer Start            |
| <code>Auto</code>      | <code>True</code>             | Verzögerter automatischer Start         |
| <code>Manual</code>    | <code>False</code>            | Manueller oder bedarfsgesteuerter Start |
| <code>Disabled</code>  | <code>False</code>            | Dienst ist deaktiviert                  |

Falls `DelayedAutoStart` auf dem untersuchten System nicht ausgegeben wird, kann der Wert gezielt in der Dienstkonfiguration der Registrierung geprüft werden:

```
[RO] Get-ItemProperty `
-Path "HKLM:\SYSTEM\CurrentControlSet\Services\<DIENSTNAME>" `
-Name DelayedAutoStart `
-ErrorAction SilentlyContinue
```

Dabei gilt üblicherweise:

| Wert                                | Bedeutung                                 |
|-------------------------------------|-------------------------------------------|
| <code>1</code>                      | Verzögerter automatischer Start aktiviert |
| <code>0</code> oder nicht vorhanden | Kein verzögerter Start daraus ableitbar   |

“ Die Registrierung darf an dieser Stelle nur gelesen werden. Eine Änderung der Startart gehört in ein freigegebenes Änderungsverfahren.

## 6. Dienststatus unter Linux mit systemd prüfen

`<DIENST>` muss durch den tatsächlichen Unit-Namen ersetzt werden, beispielsweise `nginx.service`.

### Ausführlichen Status anzeigen:

```
[RO] systemctl status "<DIENST>" --no-pager
```

### Aktuellen Laufzustand abfragen:

```
[RO] systemctl is-active "<DIENST>"
```

### Prüfen, ob die Unit für einen automatischen Start aktiviert ist:

```
[RO] systemctl is-enabled "<DIENST>"
```

### Prüfen, ob die Unit fehlgeschlagen ist:

```
[RO] systemctl is-failed "<DIENST>"
```

### Wichtige Eigenschaften kompakt anzeigen:

```
[RO] systemctl show "<DIENST>" \
--
property=Id,Names,LoadState,ActiveState,SubState,UnitFileState,MainPID,ExecMainCode,ExecMainStatus,
Result
```

### Alle fehlgeschlagenen Units anzeigen:

```
[RO] systemctl --failed --no-pager
```

### Laufende Dienste anzeigen:

```
[RO] systemctl list-units \
--type=service \
--state=running \
--no-pager
```

### Installierte Service-Unit-Dateien und Aktivierungszustände anzeigen:

```
[RO] systemctl list-unit-files --type=service --no-pager
```

“ `systemctl status` zeigt zusätzlich einige aktuelle Protokollzeilen. Für eine vollständige Protokollanalyse wird später gezielt `journalctl` verwendet.

## 7. systemd-Zustände richtig interpretieren

systemd verwendet mehrere voneinander unabhängige Zustandsangaben.

### Ladezustand:

| LoadState | Bedeutung |
|-----------|-----------|
|-----------|-----------|

|           |                                                            |
|-----------|------------------------------------------------------------|
| loaded    | Unit-Datei wurde erfolgreich geladen                       |
| not-found | Keine passende Unit-Datei gefunden                         |
| error     | Fehler beim Laden                                          |
| masked    | Unit wurde maskiert und kann nicht normal gestartet werden |

Aktivzustand:

| ActiveState  | Bedeutung                                      |
|--------------|------------------------------------------------|
| active       | Unit ist aktiv                                 |
| inactive     | Unit ist nicht aktiv                           |
| activating   | Aktivierung läuft                              |
| deactivating | Deaktivierung läuft                            |
| failed       | Aktivierung oder Ausführung ist fehlgeschlagen |
| reloading    | Konfiguration wird neu geladen                 |

Typische Aktivierungszustände:

| systemctl is-enabled | Bedeutung                                                                                  |
|----------------------|--------------------------------------------------------------------------------------------|
| enabled              | Für einen vorgesehenen automatischen Start aktiviert                                       |
| enabled-runtime      | Nur bis zum nächsten Neustart aktiviert                                                    |
| disabled             | Nicht für automatischen Start aktiviert                                                    |
| static               | Keine eigene Aktivierung über einen Installationsabschnitt; kann abhängig gestartet werden |
| indirect             | Indirekte Aktivierung über eine andere Unit vorgesehen                                     |
| masked               | Start vollständig blockiert                                                                |
| masked-runtime       | Bis zum nächsten Neustart maskiert                                                         |
| generated            | Unit wurde dynamisch erzeugt                                                               |
| transient            | Temporäre Unit                                                                             |
| not-found            | Unit wurde nicht gefunden                                                                  |

Wichtige Kombinationen:

| Laufzustand | Aktivierungszustand | Bewertung                         |
|-------------|---------------------|-----------------------------------|
| active      | enabled             | Läuft und ist dauerhaft aktiviert |

| Laufzustand | Aktivierungszustand | Bewertung                                                              |
|-------------|---------------------|------------------------------------------------------------------------|
| inactive    | enabled             | Sollte möglicherweise laufen; Start- und Protokollprüfung erforderlich |
| active      | disabled            | Wurde manuell, abhängig oder auf anderem Weg gestartet                 |
| inactive    | static              | Kann bei bedarfsgesteuerten oder abhängigen Units normal sein          |
| failed      | beliebig            | Fehlerursache und Protokolle untersuchen                               |
| inactive    | masked              | Start ist absichtlich blockiert                                        |

“ disabled bedeutet nicht automatisch „defekt“. Es bedeutet zunächst nur, dass die Unit nicht über ihre normalen Installationsverknüpfungen für den automatischen Start aktiviert ist.

## 8. systemd-Unit-Datei und Startbefehl prüfen

**Wirksame Unit-Datei einschließlich Ergänzungen anzeigen:**

```
[RO][FILE][SENS] systemctl cat "<DIENST>"
```

**Pfad der geladenen Unit-Datei anzeigen:**

```
[RO] systemctl show "<DIENST>" --property=FragmentPath
```

**Ergänzende Drop-in-Konfigurationen anzeigen:**

```
[RO] systemctl show "<DIENST>" --property=DropInPaths
```

**Startbefehl und Dienstkonto anzeigen:**

```
[RO][SENS] systemctl show "<DIENST>" \
--property=User,Group,ExecStart,WorkingDirectory,EnvironmentFiles
```

**Abhängigkeiten anzeigen:**

```
[RO] systemctl list-dependencies "<DIENST>" --no-pager
```

**Prüfen, wodurch die Unit für den Systemstart eingebunden ist:**

```
[RO] systemctl is-enabled "<DIENST>"
```

Die Ausgabe von `systemctl cat` kann enthalten:

- ursprüngliche Unit-Datei,
- lokale Überschreibungen,
- Drop-in-Dateien,
- Start- und Stopbefehle,
- Dienstkonto,
- Neustartregeln,
- Abhängigkeiten,
- Umgebungsdateien.

“ **[SENS]**: Unit-Dateien können interne Pfade, Serveradressen oder Umgebungsdateien referenzieren. Die Inhalte von geheimen Umgebungsdateien dürfen nicht ungeprüft in Tickets oder Dokumentationen kopiert werden.

**9. Dienststatus unter macOS mit launchd prüfen**

macOS unterscheidet unter anderem folgende Domains:

| Domain                        | Bedeutung                               | Beispiel                                  |
|-------------------------------|-----------------------------------------|-------------------------------------------|
| <code>system</code>           | Systemweite Dienste                     | <code>system/com.example.service</code>   |
| <code>user/&lt;UID&gt;</code> | Dienste eines Benutzerkontextes         | <code>user/501/com.example.service</code> |
| <code>gui/&lt;UID&gt;</code>  | Dienste einer grafischen Anmeldesitzung | <code>gui/501/com.example.agent</code>    |

**Benutzer-ID des aktuell angemeldeten Terminalbenutzers anzeigen:**

```
[RO] id -u
```

### Systemweiten Dienst prüfen:

```
[RO] launchctl print "system/<LABEL>"
```

### Dienst im GUI-Kontext des aktuellen Benutzers prüfen:

```
[RO] launchctl print "gui/${id -u}/<LABEL>"
```

### Gesamten Systembereich anzeigen:

```
[RO] launchctl print system
```

### GUI-Bereich des aktuellen Benutzers anzeigen:

```
[RO] launchctl print "gui/${id -u}"
```

Die Detailausgabe eines Dienstes kann unter anderem enthalten:

- Dienstlabel,
- aktuellen Zustand,
- Programmpfad,
- Startargumente,
- Prozess-ID,
- letzten Exitstatus,
- Dienstursprung auf dem Datenträger,
- Ausführungskontext,
- Startbedingungen.

“ Die Ausgabe von `launchctl print` ist für die manuelle Diagnose bestimmt. Apple garantiert ihr Format nicht als stabile Schnittstelle für Skripte.

---

## 10. Deaktivierte launchd-Dienste unter macOS prüfen

## Deaktivierungsüberschreibungen im Systembereich anzeigen:

```
[RO] launchctl print-disabled system
```

## Deaktivierungsüberschreibungen im GUI-Bereich anzeigen:

```
[RO] launchctl print-disabled "gui/$(id -u)"
```

Ein Eintrag mit `true` bedeutet, dass der Dienst in dieser Domain deaktiviert wurde. Ein Eintrag mit `false` bedeutet, dass eine Aktivierungsüberschreibung vorhanden sein kann.

### Wichtig:

- Nicht jeder Dienst erscheint in `print-disabled`.
- Ein fehlender Eintrag bedeutet nicht automatisch, dass der Dienst aktiviert ist.
- Die effektive Startentscheidung kann zusätzlich von der plist-Konfiguration und Startbedingungen abhängen.
- Aktivierungsüberschreibungen werden außerhalb der eigentlichen plist-Datei verwaltet.
- Ein launchd-Job kann aktiviert, aber momentan ohne laufenden Prozess sein.
- Bedarfsgesteuerte Dienste werden möglicherweise erst beim Eintreten einer Startbedingung ausgeführt.

## 11. launchd-Konfiguration unter macOS prüfen

Typische Speicherorte sind:

| Pfad                                       | Bedeutung                                     |
|--------------------------------------------|-----------------------------------------------|
| <code>/System/Library/LaunchDaemons</code> | Von macOS bereitgestellte systemweite Dienste |
| <code>/Library/LaunchDaemons</code>        | Zusätzlich installierte systemweite Dienste   |
| <code>/System/Library/LaunchAgents</code>  | Von macOS bereitgestellte Benutzer-Agenten    |
| <code>/Library/LaunchAgents</code>         | Zusätzlich installierte Benutzer-Agenten      |
| <code>~/Library/LaunchAgents</code>        | Agenten des jeweiligen Benutzers              |

“ Dateien unter `/System/Library` gehören zum Betriebssystem und dürfen nicht manuell verändert werden.

**Konfigurationsdatei eines bekannten Drittanbieterdienstes lesbar anzeigen:**

```
[RO][FILE][SENS] plutil -p "/Library/LaunchDaemons/<LABEL>.plist"
```

**Konfigurationsdatei auf formale plist-Fehler prüfen:**

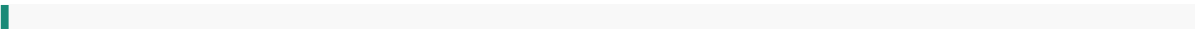
```
[RO][FILE] plutil -lint "/Library/LaunchDaemons/<LABEL>.plist"
```

**Benutzerbezogenen Agenten anzeigen:**

```
[RO][FILE][SENS] plutil -p "$HOME/Library/LaunchAgents/<LABEL>.plist"
```

Typische Schlüssel einer launchd-plist:

| Schlüssel             | Bedeutung                                                    |
|-----------------------|--------------------------------------------------------------|
| Label                 | Eindeutige Dienstbezeichnung                                 |
| Program               | Auszuführendes Programm                                      |
| ProgramArguments      | Programm und Argumente                                       |
| RunAtLoad             | Start beim Laden der Jobdefinition                           |
| KeepAlive             | Bedingungen für Ausführung beziehungsweise Neustart          |
| StartInterval         | Zeitgesteuerter Start in Sekundenabständen                   |
| StartCalendarInterval | Kalenderbasierter Start                                      |
| WatchPaths            | Start bei Änderungen überwachter Pfade                       |
| QueueDirectories      | Start, wenn ein überwachtes Verzeichnis nicht leer ist       |
| UserName              | Benutzerkonto eines systemweiten Daemons                     |
| GroupName             | Gruppenkonto eines systemweiten Daemons                      |
| WorkingDirectory      | Arbeitsverzeichnis                                           |
| StandardOutPath       | Datei für Standardausgabe                                    |
| StandardErrorPath     | Datei für Fehlerausgabe                                      |
| Disabled              | Standardvorgabe zur Deaktivierung; kann überschrieben werden |



`RunAtLoad`, `KeepAlive` und andere Startbedingungen sind nicht direkt mit den Windows-Startarten gleichzusetzen. launchd ist stark auf bedarfs- und ereignisgesteuertes Starten ausgelegt.

## 12. Letzten Exitstatus und wiederholte Starts erkennen

| Betriebssystem | Prüfung                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------|
| Windows        | Dienststatus, Prozess-ID, Exitcode und Ereignisprotokolle                                            |
| Linux          | <code>Result</code> <code>ExecMainCode</code> <code>ExecMainStatus</code> Neustartzähler und Journal |
| macOS          | <code>state</code> <code>pid</code> <code>last exit code</code> und Unified Logging                  |

### Windows:

```
[RO] Get-CimInstance Win32_Service -Filter "Name='<DIENSTNAME>'" |  
Select-Object Name, State, ProcessId, ExitCode, ServiceSpecificExitCode
```

### Linux:

```
[RO] systemctl show "<DIENST>" \  
--property=ActiveState,SubState,Result,MainPID,ExecMainCode,ExecMainStatus,NRestarts
```

### macOS:

```
[RO] launchctl print "system/<LABEL>"
```

Bei macOS sind insbesondere folgende Angaben relevant:

```
state  
pid  
last exit code  
runs  
reason
```

Die tatsächlich vorhandenen Felder können abhängig vom Dienst und von der macOS-Version abweichen.

**Verdächtige Anzeichen:**

- Prozess-ID wechselt in kurzen Abständen,
- Neustartzähler steigt,
- Dienststatus springt zwischen Start und Stopp,
- letzter Exitcode ist ungleich null,
- Dienst wird durch ein Signal beendet,
- Startgrenze oder Neustartbegrenzung wird erreicht,
- Dienst läuft nur wenige Sekunden,
- abhängiger Dienst ist nicht verfügbar.

“ Ein Exitcode ist zunächst nur ein technischer Hinweis. Seine genaue Bedeutung muss anhand der Dienst-, Hersteller- oder Betriebssystemdokumentation bestimmt werden.

**13. Status vor einer Änderung dokumentieren**

Vor einem Start, Stopp oder Neustart sollten mindestens folgende Informationen gesichert werden:

| Information                 | Zweck                                      |
|-----------------------------|--------------------------------------------|
| Datum, Uhrzeit und Zeitzone | Zuordnung zu Protokolleinträgen            |
| Dienstname und Anzeigename  | Eindeutige Identifikation                  |
| Aktueller Status            | Ausgangszustand                            |
| Konfigurierte Startart      | Sollverhalten                              |
| Prozess-ID                  | Zuordnung zum Prozess                      |
| Dienstkonto                 | Berechtigungsprüfung                       |
| Programmpfad und Argumente  | Konfigurationsprüfung                      |
| Exitcode                    | Hinweis auf Beendigungsursache             |
| Abhängigkeiten              | Auswirkungsanalyse                         |
| Neustartregeln              | Erkennung automatischer Wiederanläufe      |
| Letzte Protokollmeldungen   | Erhaltung flüchtiger Diagnoseinformationen |

| Information         | Zweck                               |
|---------------------|-------------------------------------|
| Aktive Verbindungen | Bewertung möglicher Unterbrechungen |

Kompakte Dokumentationsvorlage:

Zeitpunkt:

Server:

Betriebssystem:

Dienstname:

Anzeigename beziehungsweise Label:

Dienststatus:

Startart beziehungsweise Aktivierung:

Prozess-ID:

Dienstkonto:

Programmpfad:

Letzter Exitcode:

Neustartzähler:

Abhängigkeiten:

Relevante Protokollmeldung:

Bewertung:

Nächster Prüfschritt:

14. Dienstzustand verändern - nur nach Freigabe

Die folgenden Befehle sind **keine reinen Diagnosebefehle**. Sie verändern den Zustand und können Benutzer, Verbindungen oder Datenverarbeitung beeinträchtigen.

| Aufgabe            | Windows PowerShell                                                                        | Linux mit systemd                                                            | macOS mit launchd                                                                       |
|--------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| Dienst starten     | <code>[CHANGE][PRIV] Start-Service -Name "&lt;DIENSTNAME&gt;"</code>                      | <code>[CHANGE][PRIV] sudo systemctl start "&lt;DIENST&gt;"</code>            | <code>[CHANGE][PRIV] sudo launchctl kickstart "system/&lt;LABEL&gt;"</code>             |
| Dienst stoppen     | <code>[CHANGE][DISRUPT][PRIV] Stop-Service -Name "&lt;DIENSTNAME&gt;"</code>              | <code>[CHANGE][DISRUPT][PRIV] sudo systemctl stop "&lt;DIENST&gt;"</code>    | <code>[CHANGE][DISRUPT][PRIV] sudo launchctl kill SIGTERM "system/&lt;LABEL&gt;"</code> |
| Dienst neu starten | <code>[CHANGE][DISRUPT][PRIV] Restart-Service -Name "&lt;DIENSTNAME&gt;"</code>           | <code>[CHANGE][DISRUPT][PRIV] sudo systemctl restart "&lt;DIENST&gt;"</code> | <code>[CHANGE][DISRUPT][PRIV] sudo launchctl kickstart -k "system/&lt;LABEL&gt;"</code> |
| Aktivieren         | <code>[CHANGE][PRIV] Set-Service -Name "&lt;DIENSTNAME&gt;" -StartupType Automatic</code> | <code>[CHANGE][PRIV] sudo systemctl enable "&lt;DIENST&gt;"</code>           | <code>[CHANGE][PRIV] sudo launchctl enable "system/&lt;LABEL&gt;"</code>                |

| Aufgabe      | Windows PowerShell                                                                                | Linux mit systemd                                                   | macOS mit launchd                                                         |
|--------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------------|
| Deaktivieren | <code>[CHANGE][DISRUPT][PRIV] Set-Service -Name "&lt;DIENSTNAME&gt;" -StartupType Disabled</code> | <code>[CHANGE][PRIV] sudo systemctl disable "&lt;DIENST&gt;"</code> | <code>[CHANGE][PRIV] sudo launchctl disable "system/&lt;LABEL&gt;"</code> |

## Wichtige Unterschiede:

- `systemctl enable` startet einen Dienst normalerweise nicht sofort.
- `systemctl start` aktiviert ihn nicht automatisch für spätere Systemstarts.
- `systemctl enable --now` würde beide Vorgänge kombinieren und ist deshalb eine weitergehende Änderung.
- `launchctl kickstart -k` beendet eine bereits laufende Instanz und startet sie neu.
- `launchctl disable` speichert eine Deaktivierungsüberschreibung, die Neustarts überdauern kann.
- Das Senden von `SIGTERM` beendet den aktuellen Prozess. launchd kann ihn abhängig von der Konfiguration erneut starten.
- Das Ändern der Startart beseitigt nicht die Ursache eines Startfehlers.

“ Diese Befehle sollten erst nach Sicherung des Ausgangszustands, Prüfung der Auswirkungen und betrieblicher Freigabe eingesetzt werden.

## 15. Typische Fehlinterpretationen

| Fehlinterpretation                                                                 | Tatsächliche Bedeutung                                                   |
|------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| „Der Dienst ist beendet, also ist er defekt.“                                      | Ein manueller oder bedarfsgesteuerter Dienst kann regulär beendet sein   |
| „Der Dienst ist aktiviert, also läuft er.“                                         | Aktivierung und aktueller Laufzustand sind getrennte Eigenschaften       |
| „Der Prozess läuft, also funktioniert der Dienst.“                                 | Port, Protokoll und Anwendungsfunktion können trotzdem fehlerhaft sein   |
| „systemd zeigt <code>static</code> , also ist die Unit deaktiviert.“               | Die Unit kann durch Abhängigkeiten oder andere Auslöser gestartet werden |
| „systemd zeigt <code>disabled</code> , also kann die Unit nicht gestartet werden.“ | Ein deaktivierter Dienst kann weiterhin manuell gestartet werden         |
| „systemd zeigt <code>masked</code> , das ist dasselbe wie <code>disabled</code> “  | Eine maskierte Unit ist zusätzlich gegen normalen Start blockiert        |
| „launchd zeigt keine PID, also ist der Dienst defekt.“                             | Ein bedarfsgesteuerter Job kann momentan ohne Prozess sein               |

| Fehlinterpretation                            | Tatsächliche Bedeutung                                                  |
|-----------------------------------------------|-------------------------------------------------------------------------|
| „Exitcode 0 beweist vollständige Funktion.“   | Er zeigt nur eine erfolgreiche Beendigung nach Definition des Programms |
| „Ein Neustart hat das Problem gelöst.“        | Er kann nur das sichtbare Fehlerbild vorübergehend beseitigt haben      |
| „Automatischer Neustart ist immer erwünscht.“ | Er kann Abstürze verdecken oder eine Neustartschleife erzeugen          |

## 16. Checkliste zur Status- und Startartprüfung

- ☐ Internen Dienstnamen beziehungsweise launchd-Label ermittelt
- ☐ Anzeigenamen und technischen Namen unterschieden
- ☐ Richtigen Server und richtige Umgebung bestätigt
- ☐ Aktuellen Dienststatus erfasst
- ☐ Startart beziehungsweise Aktivierungszustand geprüft
- ☐ Bedarfsgesteuerten Betrieb berücksichtigt
- ☐ Deaktivierung oder Maskierung geprüft
- ☐ Prozess-ID erfasst
- ☐ Dienstkonto beziehungsweise Ausführungskontext geprüft
- ☐ Programmpfad und Startargumente geprüft
- ☐ Exitcode und dienstspezifischen Exitcode erfasst
- ☐ Neustartzähler beziehungsweise wiederholte Starts geprüft
- ☐ Abhängigkeiten aufgenommen
- ☐ Ausgangszustand vor Änderungen dokumentiert
- ☐ Noch keine ungeprüfte Zustandsänderung durchgeführt
- ☐ Nächsten Diagnoseschritt festgelegt

## Bewertung des Ergebnisses

| Ergebnis                                           | Nächster Schritt                                         |
|----------------------------------------------------|----------------------------------------------------------|
| Dienst ist nicht registriert oder nicht auffindbar | Installation, Produktdokumentation und Zielsystem prüfen |
| Dienst ist deaktiviert oder maskiert               | Grund der Deaktivierung und Änderungsverlauf klären      |
| Dienst ist beendet, sollte aber laufen             | Startfehler, Exitcode und Protokolle untersuchen         |
| Dienst hängt im Übergangszustand                   | Prozess, Abhängigkeiten und Protokolle prüfen            |

| Ergebnis                                  | Nächster Schritt                                         |
|-------------------------------------------|----------------------------------------------------------|
| Dienst startet und beendet sich wieder    | Exitcode, Neustartregeln und Anwendungsprotokolle prüfen |
| Dienst läuft stabil                       | Prozess, Listener, Port und Anwendungsfunktion prüfen    |
| Dienst läuft unter falschem Konto         | Konfiguration und benötigte Berechtigungen untersuchen   |
| Startart entspricht nicht dem Sollzustand | Sollkonfiguration und letzte Änderungen prüfen           |
| Zustand ist unklar                        | Keine Änderung durchführen; weitere Beweise sammeln      |

## Merksatz

“ „Aktiviert“ beschreibt die Startkonfiguration. „Läuft“ beschreibt den aktuellen Zustand. „Funktioniert“ muss durch einen eigenen Anwendungstest bewiesen werden.

## Weiterführende Quellen

- [Microsoft Learn – Get-Service](#)
- [Microsoft Learn – Get-CimInstance](#)
- [Microsoft Learn – Win32\\_Service](#)
- [Microsoft Learn – sc.exe query](#)
- [Microsoft Learn – sc.exe qc](#)
- [systemd – systemctl](#)
- [systemd – systemd.service](#)
- [systemd – systemd.unit](#)
- [Apple – launchctl-Handbuchseite](#)
- [Apple – launchd.plist-Handbuchseite](#)