DELL Technologies

WHITEPAPER

Whitepaper zur Korrektur mit SupportAssist for Business PCs

Übersicht, Nutzung und Sicherheitsinformationen

Autoren: Gus Chavira und Sven Riebe

Mitwirkende: Rucha Spare, Laura Trammell, Ravi B, Niraj Shah, Vikas Sharma

Dell.com/SupportAssist

INHALTSVERZEICHNIS

Einführung

SupportAssist-Korrekturskripte bieten IT-AdministratorInnen eine rationalisierte, flexible Möglichkeit, den Zustand der PC-Flotte zu verbessern und aufrechtzuerhalten. Dieses Whitepaper bietet eine Übersicht über Korrekturskripte und nutzerdefinierte Korrekturworkflows. Es bietet einen transparenten Einblick in die Skriptprotokollierung und -ausgabe, beschreibt, wie Skripte sicher verwaltet werden, und enthält praktische Anweisungen für erweiterte benutzerdefinierte Skriptworkflows

Übersicht über die Korrektur mit SupportAssist

Die Wiederherstellungsregeln sind eine Funktion innerhalb des Dashboards zum Verbinden und Verwalten von PCs in TechDirect, das KundInnen zur Verfügung steht, die SupportAssist for Business PCs verwenden. Diese Funktion deckt die Korrekturplattform ab, die SupportAssist-Versionen ab 4.5.2.x verwendet. Die Korrekturfunktion unterstützt von Dell erstellte und nutzerdefinierte Korrekturskripte. Im Abschnitt für Korrekturen gibt es verschiedene Arten von Skripten, wie unten aufgeführt:

- **Vollständige End-to-End-Korrekturen:** Diese Skripte erkennen und beheben Probleme automatisch, sobald sie erkannt werden. Einige Skripte ermöglichen nur die Erkennung, sodass AdministratorInnen zunächst die Informationen überprüfen und dann flexibel entscheiden können, ob die Korrektur fortgesetzt werden soll oder nicht.
- **Nur Erkennung:** Diese Skripte liefern wertvolle Erkenntnisse, indem sie potenzielle Probleme in der gesamten Flotte erkennen. Sie verzichten bewusst auf Korrekturfunktionen, sodass AdministratorInnen die Flexibilität haben, die Informationen zuerst zu überprüfen, eine Korrektur zu wählen oder nach Bedarf andere Korrekturbereitstellungsoptionen zu erkunden.
- **Optimierung:** Diese Skripte wurden entwickelt, um Einstellungsänderungen zu implementieren, z. B. das Ändern einer BIOS- oder Betriebssystemkonfiguration oder die Installation von Software, die die Performance und Effizienz eines Endpunkts verbessert.

Berechtigungsanforderungen für Korrekturskripte

Korrekturskripte erfordern Berechtigungen für **ProSupport Plus oder ProSupport Flex for PCs**, um auf einem Clientgerät ausgeführt werden zu können. IT-AdministratorInnen können diese Skripte zwar in TechDirect auf eine gesamte Flotte anwenden, sie werden jedoch nur auf PCs mit einer aktiven ProSupport Plus- oder ProSupport Flex-Berechtigung ausgeführt. Geräte mit abgelaufenen Berechtigungen können die Skripte unabhängig von ihrem vorherigen Berechtigungsstatus nicht ausführen. AdministratorInnen müssen nicht nachverfolgen, welche Geräte berechtigt sind, da die Plattform die Berechtigung automatisch während der Planung überprüft.

Warum sollten Sie Korrekturskripte ausführen?

Die Korrekturplattform von Dell bietet eine Reihe von Funktionen, darunter kundenspezifische Workflows mit optionaler, vom Kunden bereitgestellter Zertifikatskriptsignierung oder nicht zertifikatsignierten Skripten, telemetriegesteuerte Korrekturen und verbesserte Korrekturausgaben. Die Nutzung der umfangreichen und kontinuierlich wachsenden Bibliothek mit von Dell verfassten Skripten hilft bei der Optimierung und Sicherung von PCs, liefert wertvolle Einblicke in Informationen auf Flottenebene und ermöglicht den Export von Daten zur Unterstützung von nutzerdefinierten Diagrammen oder Dashboard-Berichten. Darüber hinaus kann mit nutzerdefinierten Workflows eine Skriptbibliothek hinzugefügt werden, um Funktionen zu berücksichtigen, die nicht von den von Dell verfassten Skripten abgedeckt sind, und so ein sicheres und skalierbares Management der PC-Flotte zu gewährleisten.

Korrekturregeln und Sicherheitsinformationen

In diesem Abschnitt wird beschrieben, wie Skripte im Ruhezustand, während der Übertragung und vor der Ausführung mithilfe der Korrekturfunktionen im Dashboard zum Verbinden und Verwalten von PCs in TechDirect sicher gemanagt werden.

Vor dem Hochladen auf die Korrekturplattform werden alle von Dell verfassten Korrekturskripte mit Dell Zertifikaten signiert und umfangreichen Tests und Validierungen unterzogen, um sicherzustellen, dass sie wie vorgesehen funktionieren, ohne unerwartete Ergebnisse zu erzielen. Dies dient als Grundlage für die Überprüfung der Authentizität des Skripts vor der Ausführung. Wird ein Skript auf dem Endpunkt geändert oder ersetzt, schlägt die Validierung der Zertifikatsignatur fehl und SupportAssist blockiert die Ausführung des Skripts. Dies verhindert die Ausführung von unbefugtem oder potenziell schädlichem Code.

Für nutzerdefinierte Workflowskripte wird ein anderer Prozess befolgt. Wenn Kunden ihre eigenen Skripte hochladen, akzeptiert Dell sowohl nicht signierte Skripte als auch Skripte, die mit einem Kundenzertifikat signiert sind. Die Integrität dieser Skripte bleibt sowohl während des Transports zu PCs und als auch während der Speicherung erhalten.

Dell empfiehlt, Skripte vor einer breiteren Bereitstellung auf einer bestimmten Gruppe von PCs zu testen. Das Dashboard zum Verbinden und Verwalten von PCs von TechDirect unterstützt die Erstellung von Standorten und Gruppen, sodass Kunden sowohl von Dell erstellte als auch benutzerdefinierte Skripte auf Testmaschinen validieren können. Alle Informationen in der Korrekturkonsole sind innerhalb der Mandantengrenzen in TechDirect gesichert und nur für NutzerInnen mit den entsprechenden Rollen zugänglich, die vom Mandantenadministrator zugewiesen wurden. Die Ergebnisse können auch zur weiteren Analyse in eine CSV-Datei exportiert werden.

Umfassende Informationen zur Sicherheit innerhalb der SupportAssist-Umgebung finden Sie im Whitepaper zur Sicherheit von SupportAssist for Business PCs.

Protokollierung und Transparenz in von Dell erstellten Skripten

Zur Fehlerbehebung und Transparenz werden von Dell erstellte Skripte mithilfe der Microsoft Windows-Ereignisprotokollierung protokolliert. Diese Ereignisse können auf zusätzliche Details überprüft und auch mit Protokollierungs- und Dashboarding-Tools verwendet werden, um Protokollanalysen durchzuführen oder den Flottenstatus über Widgets in einem Dashboard anzuzeigen. Die Protokolle werden nicht von Dell erfasst und nur lokal auf dem Gerät gespeichert.

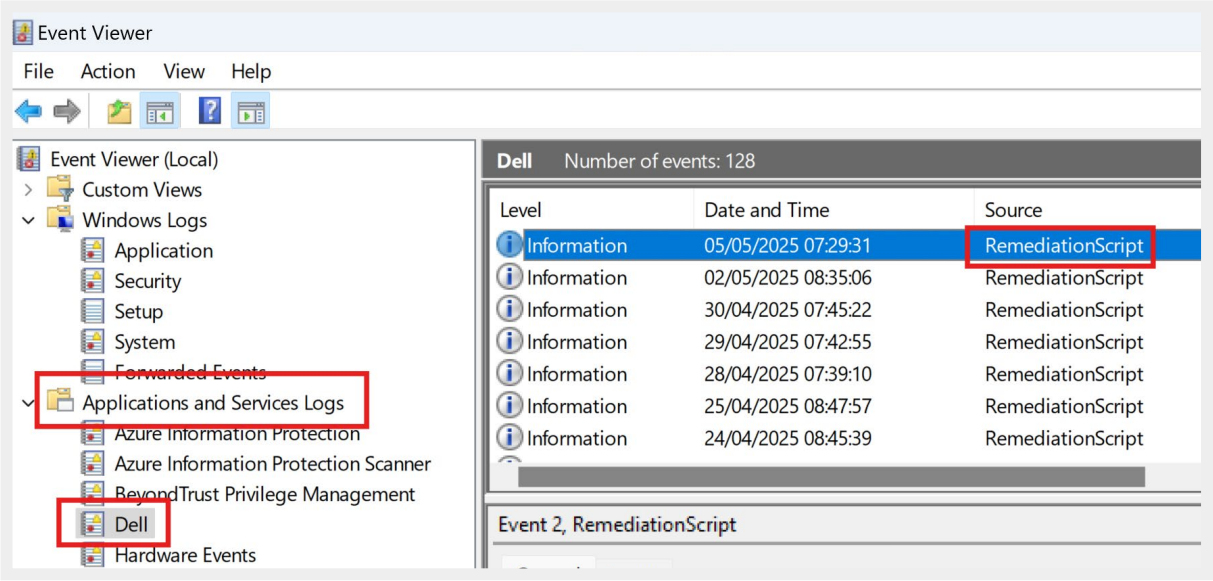
Ereignistypen, die in Microsoft-Ereignisprotokollen erfasst werden

- Telemetrieergebnisse
- Skriptausführungen
- Unteraufgaben aus Skriptausführungen

Zugriff auf Ereignisprotokolle

Um diese Protokolle anzuzeigen, öffnen Sie die Microsoft-Ereignisanzeige mit Administratorrechten. Ohne die entsprechenden Berechtigungen sind bestimmte Protokolle möglicherweise nicht zugänglich, was die Sichtbarkeit kritischer Ereignisdaten einschränkt.

Navigieren Sie zum Abschnitt **Anwendungs- und Dienstprotokolle** und suchen Sie Einträge, bei denen der Protokollname auf **„Dell“** und der Quellname auf **„RemediationScript“** festgelegt ist.



Windows-Ereignisprotokollierung und Ereignisdetails

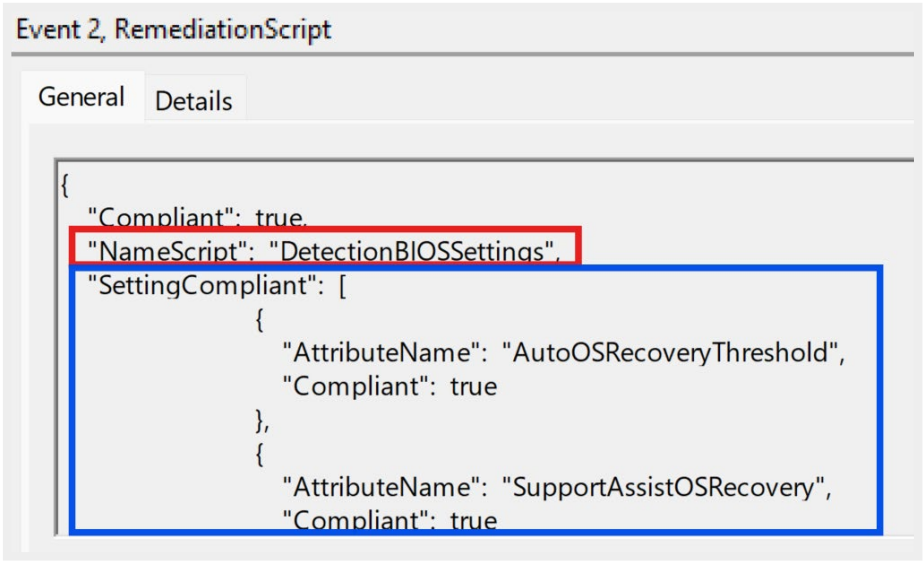
Um relevante Protokolle effizienter zu identifizieren, werden die folgenden Quellen verwendet:

- **RemediationScript:** In diesem Quellprotokollskript werden Telemetriedaten und Ressourcendetails ausgeführt, z. B. Informationen zur Dockingstation-Firmware.
- **RemediationFunction:** Dell nutzt Module, um von Dell erstellte Skripte einfacher zu pflegen. Diese Quelle erfasst die Ergebnisse von Modulen, wenn sie von einem Skript aufgerufen werden.
- **RemediationInstall:** Bestimmte Skripte erfordern die Dell Client Management-Software für die Ausführung. Wenn die Software nicht auf dem Clientgerät vorhanden ist, wird sie vom Skript installiert und der Prozess wird unter dieser Quelle protokolliert.
- **RemediationTranscript:** Dieses Laufzeitprotokoll enthält detaillierte Informationen, einschließlich Nutzerkontext, PowerShell-Version, Skriptprotokoll und alle während der Ausführung aufgetretenen Fehler.

Ebene und Ereignis-ID:
Um den Prozess der Identifizierung von Fehlern in Protokollen zu vereinfachen, werden Ebene und Ereignis-ID verwendet.

Quelle	Ebene	Ereignis-ID
RemediationScript	SuccessAudit	0
	Fehler	1
	Informationen	2
	Warnung	3
RemediationFunction	SuccessAudit	10
	Fehler	11
	Informationen	12
	Warnung	13
RemediationInstall	SuccessAudit	20
	Fehler	21
	Informationen	22
	Warnung	23
RemediationTranscript	Fehler	1
	Informationen	2
	Warnung	2

Wenn Sie mehrere Dell Workflows ausführen, überprüfen Sie den **Nachrichtentext** in den Protokollen, um das Skript zu identifizieren, das für die Generierung des Eintrags verantwortlich ist. Das rote Feld hebt den Skriptnamen hervor, während das blaue Feld ein Beispiel für die zusätzlichen Daten enthält, die in diesen Protokollen verfügbar sind.



Um diese Informationen mithilfe von PowerShell im Detail abzurufen, führen Sie die folgenden Schritte aus:

```
Get-WinEvent -FilterHashtable @{ LogName=”Dell”;
ProviderName=”RemediationScript”;
ID=2 }-MaxEvents 10 -ErrorAction Stop
```

```
TimeCreated                     Id LevelDisplayName Message
-----------                     -- --------------- -------
05/05/2025 07:29:31              2 Information      {…
02/05/2025 08:35:06              2 Information      {…
30/04/2025 07:45:22              2 Information      {…
29/04/2025 07:42:55              2 Information      {…
28/04/2025 07:39:10              2 Information      {…
```

1. Der Nachrichtentext dieser Ereignisprotokolle kann verwendet werden, um eine Hash-Tabelle für zusätzliche Aktionen zu erstellen:

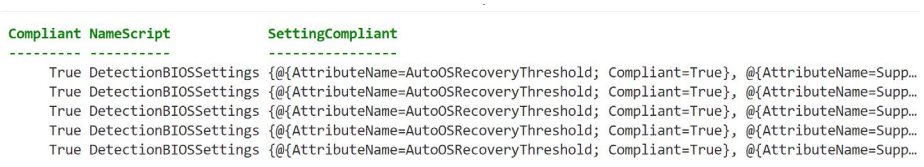
```
$Events = Get-WinEvent -FilterHashtable @{ LogName=”Dell”;
ProviderName=”RemediationScript”;
ID=2 } -MaxEvents 10 -ErrorAction Stop
```

2. Konvertieren Sie den Nachrichtentext in eine Hash-Tabelle:

```
$HashTable = $events.message | ConvertFrom-Json
```

3. Überprüfen Sie die Hash-Tabelle:

```
$HashTable
```



```
Compliant NameScript            SettingCompliant
--------- ----------            ----------------
     True DetectionBIOSSettings {@{AttributeName=AutoOSRecoveryThreshold; Compliant=True}, @{AttributeName=Supp…
     True DetectionBIOSSettings {@{AttributeName=AutoOSRecoveryThreshold; Compliant=True}, @{AttributeName=Supp…
     True DetectionBIOSSettings {@{AttributeName=AutoOSRecoveryThreshold; Compliant=True}, @{AttributeName=Supp…
     True DetectionBIOSSettings {@{AttributeName=AutoOSRecoveryThreshold; Compliant=True}, @{AttributeName=Supp…
     True DetectionBIOSSettings {@{AttributeName=AutoOSRecoveryThreshold; Compliant=True}, @{AttributeName=Supp…
```

Dieser Ansatz vereinfacht die Fehlerbehebung für von Dell erstellte Skripts und bietet die Möglichkeit, diese Protokolle für zusätzliche Überwachungszwecke zu nutzen.

Übersicht über die von Dell erstellte Skriptausgabe in TechDirect

Wenn von Dell erstellte Skripte ausgeführt werden, wird ihre Geräteausgabe in TechDirect angezeigt. Diese Ausgabe ist in zwei Hauptabschnitte unterteilt, einen für die Regelausgabe und einen für den Korrekturstatus. Kunden können für Dell Korrekturskripte und nutzerdefinierte Workflowskripte auch den Status auf Workflowebene anzeigen, wodurch eine umfassende Transparenz gewährleistet ist. Im Folgenden finden Sie eine detaillierte Beschreibung dieser Abschnitte:

Regelausgabe

Der Abschnitt für die Regelausgabe enthält Details zu den Erfolgs- oder Fehlerkriterien des Skripts. Er umfasst in der Regel erweiterte Statusangaben. Wenn beispielsweise ein BitLocker-Skript ausgeführt wird, kann in diesem Abschnitt angezeigt werden, ob ein Gerät verschlüsselt ist oder nicht, sowie der spezifische Verschlüsselungsstatus.

Zu den wichtigsten Funktionen dieses Abschnitts gehören:

- **Sortierbarkeit:** Organisieren Sie die Informationen in dieser Spalte, um sich auf die relevantesten Informationen zu konzentrieren.
- **Datenexport:** Verwenden Sie die Funktion zum Exportieren in CSV, um maßgeschneiderte Berichte oder Dashboards zu erstellen

Korrekturstatus

Im Abschnitt für den Korrekturstatus wird der Sendestatus der Skriptergebnisse angezeigt, um zu bestätigen, ob das Skript erfolgreich vom Gerät empfangen und verarbeitet wurde.

So interpretieren Sie die Statusanzeigen:

- **Grün** steht für „Erfolg“ und wird von einer Erfolgsmeldung begleitet.
- **Rot** zeigt an, dass möglicherweise weitere Maßnahmen erforderlich sind, und wird von einer Fehlermeldung begleitet. Dieser Status bedeutet nicht immer, dass das Skript fehlgeschlagen ist. Es wird jedoch empfohlen, die Details zu überprüfen, um weitere Klarheit zu gewinnen.

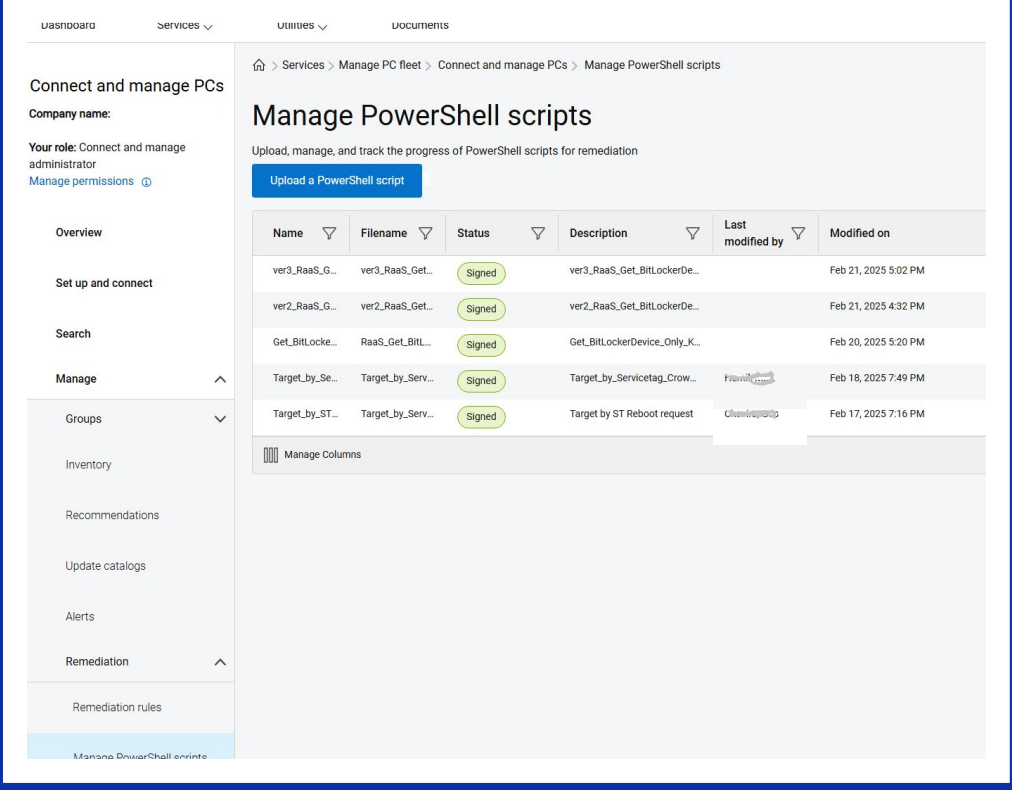
Die Interpretation dieser Status kann je nach Zweck des Skripts variieren, je nachdem, ob es für die vollständige Behebung, Optimierung oder Informationsanzeige konzipiert ist.

Übersicht über den nutzerdefinierten Workflow für die Korrektur

Skriptsignierungs- und Uploadprozess

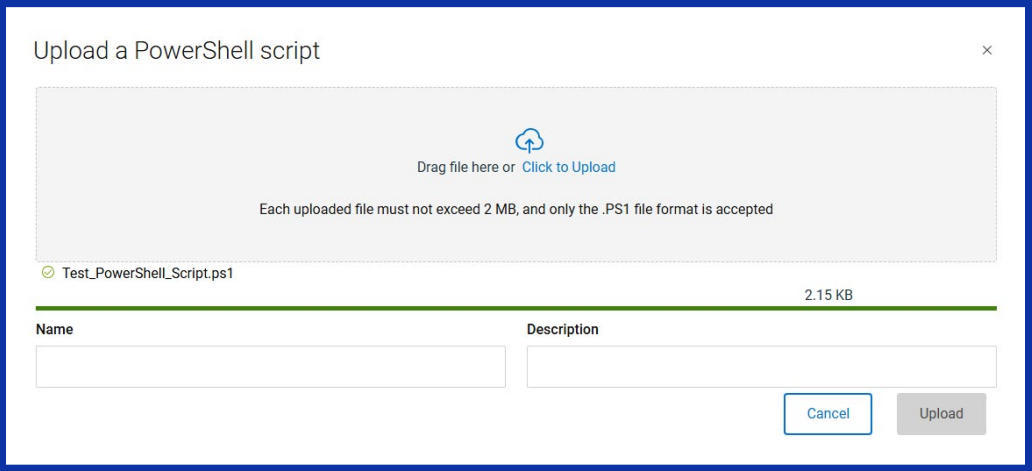
Wenn ein nutzerdefiniertes Skript auf die Korrekturplattform hochgeladen wird, kann es entweder mit einem vom Kunden bereitgestellten Zertifikat signiert oder als nicht signiertes Skript hochgeladen werden. Unabhängig vom Skripttyp wird dessen Integrität während der Übertragung zum PC, im Ruhezustand und während der gesamten Ausführung gewahrt. Diese Maßnahme stellt sicher, dass das Skript in allen Phasen sicher bleibt:

Es gibt zwei Methoden zum Hochladen nutzerdefinierter Skripte:



1. **Über den Abschnitt zum Verwalten von PowerShell-Skripten**
   - Navigieren Sie zum Abschnitt **Remediation** im Dashboard **Connect and manage PCs** in TechDirect.
   - Wählen Sie **Manage PowerShell** und ziehen Sie das Skript per Drag-and-Drop, um es hochzuladen.

2. **Inline-Upload im Workflow-Canvas**
   - Laden Sie das Skript beim Erstellen eines benutzerdefinierten Workflows zur Fehlerbehebung direkt in die **Workflow Canvas**-Oberfläche.



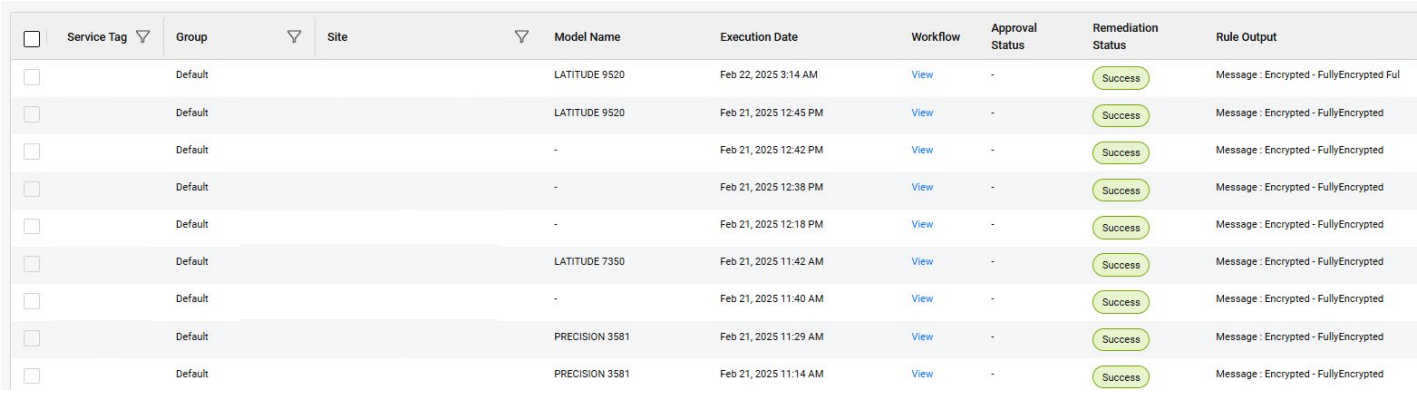
Bei beiden Methoden müssen Sie das Skript per Drag-and-Drop verschieben, benennen und eine Beschreibung hinzufügen.

Generieren einer formatierten Ausgabe für die Korrekturplattform

Vor dem Hochladen eines nutzerdefinierten PowerShell-Skripts müssen wichtige Ausgabefunktionen konfiguriert werden. Diese Ausgaben liefern nach der Ausführung des Skripts für die Flotte verwertbare Einblicke in die TechDirect-Benutzeroberfläche.

Exit-Werte für die Statusanzeige

- Verwenden Sie **Exit-Codes**, um den Erfolg oder Fehler eines Skripts anzuzeigen.
  - Exit-Code 0 steht für **Erfolg** (in Grün angezeigt).
  - Exit-Code 1 steht für **Fehler** (in Rot angezeigt).
- Always ensure the script logic assigns clear success or failure conditions at every exit point.



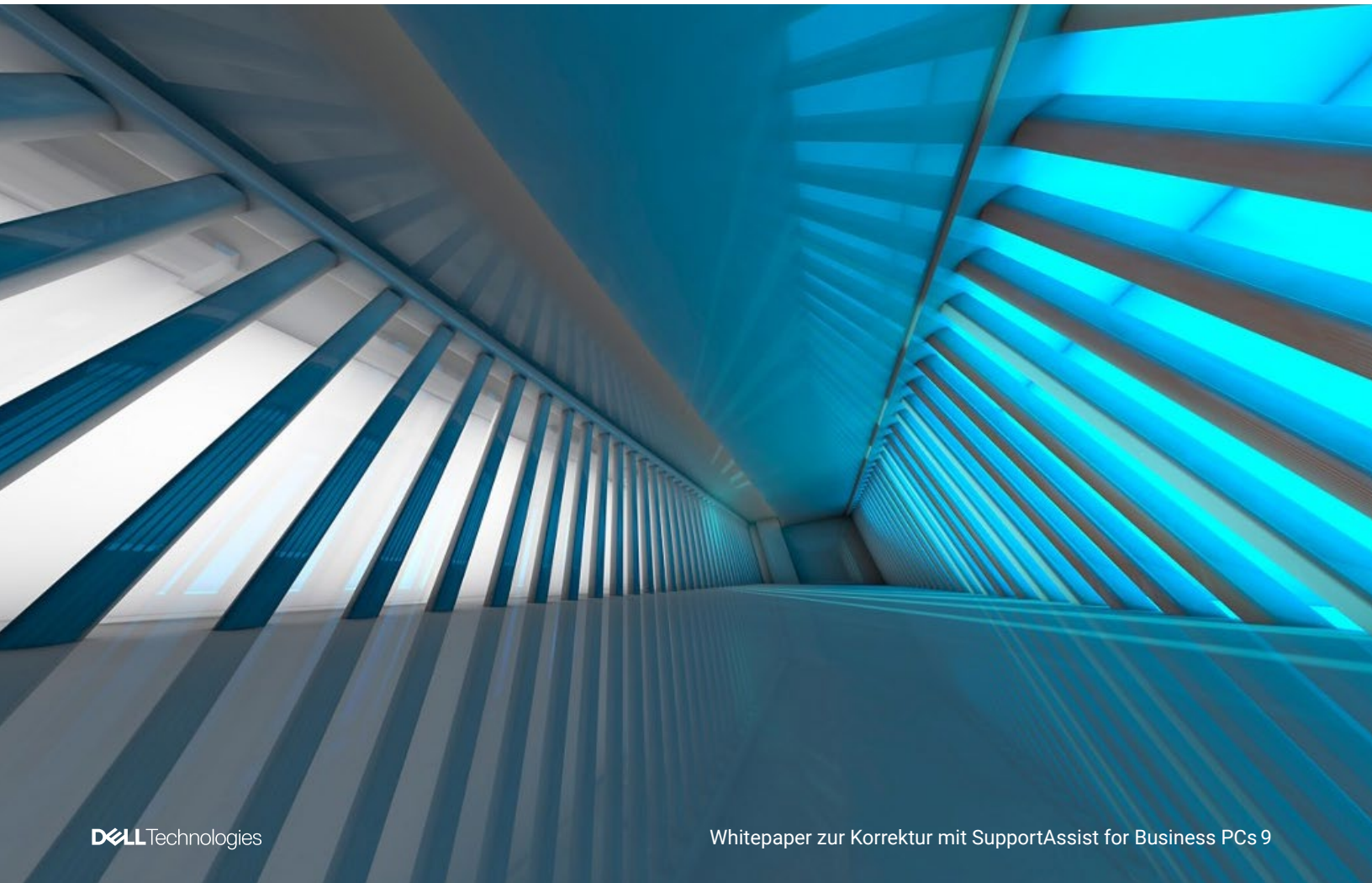
Service Tag	Group	Site	Model Name	Execution Date	Workflow	Approval Status	Remediation Status	Rule Output
	Default		LATITUDE 9520	Feb 22, 2025 3:14 AM	View	-	Success	Message : Encrypted - FullyEncrypted Ful
	Default		LATITUDE 9520	Feb 21, 2025 12:45 PM	View	-	Success	Message : Encrypted - FullyEncrypted
	Default		-	Feb 21, 2025 12:42 PM	View	-	Success	Message : Encrypted - FullyEncrypted
	Default		-	Feb 21, 2025 12:38 PM	View	-	Success	Message : Encrypted - FullyEncrypted
	Default		-	Feb 21, 2025 12:18 PM	View	-	Success	Message : Encrypted - FullyEncrypted
	Default		LATITUDE 7350	Feb 21, 2025 11:42 AM	View	-	Success	Message : Encrypted - FullyEncrypted
	Default		-	Feb 21, 2025 11:40 AM	View	-	Success	Message : Encrypted - FullyEncrypted
	Default		PRECISION 3581	Feb 21, 2025 11:29 AM	View	-	Success	Message : Encrypted - FullyEncrypted
	Default		PRECISION 3581	Feb 21, 2025 11:14 AM	View	-	Success	Message : Encrypted - FullyEncrypted

Spalte für die Regelausgabe

- Füllen Sie die Spalte **Remediation Status** mit sortierbaren, prägnanten Statusmeldungen (maximal 40 Zeichen). Fügen Sie Kontextinformationen mit einem „|“ (senkrechten Strich) als Trennzeichen um den Ausgabetext ein. Dieses Trennzeichen markiert den Text, der in der Spalte für die Regelausgabe angezeigt werden soll. Beispiel: Write-Host „|Message: Encrypted - $BLstatus|“
- Dies führt zu folgender Ausgabe in der Spalte **Rule Output**:

Rule Output
Message : Encrypted - FullyEncrypted
Message : Encrypted - FullyEncrypted
Message : Encrypted - FullyEncrypted

Hinweis: Wenn mehrere Ausgaben definiert sind, wird nur die letzte Ausgabe angezeigt.

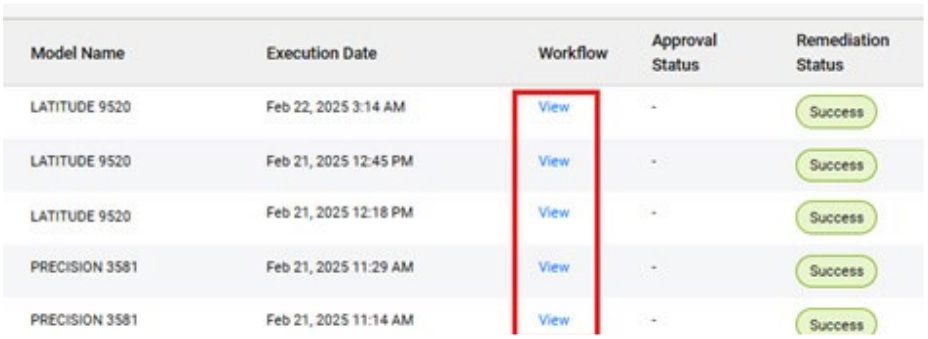


Detaillierte Ausgabe für den Abschnitt „Workflow“

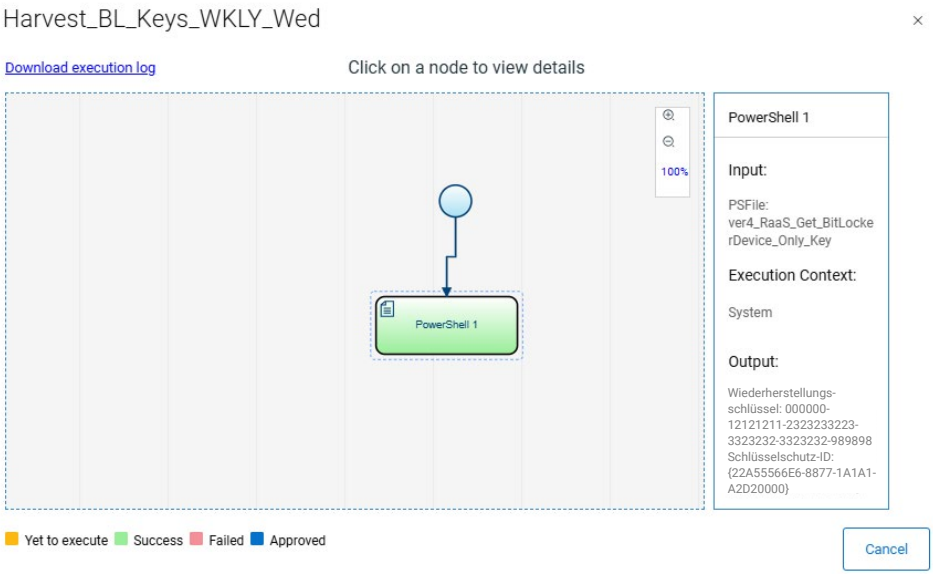
- Für längere oder detailliertere Ausgaben verwenden Sie den Abschnitt „Workflow“ in der Benutzeroberfläche. Diese Ausgabe wird durch Klicken auf den Abschnitt zum Anzeigen der Workflow-Ergebnisse aufgerufen.
- Verwenden Sie die Trennzeichen „~~“ (doppelte Tilde) um den Text, um ihn in diesen Abschnitt zu leiten. Beispiel:

  Write-Host „~~Recovery key: $RecoveryKey1 Key Protector ID: $KeyProtectorID1~~“

  Dadurch werden detaillierte Informationen angezeigt, wenn Sie das Ausgabefeld des Abschnitts „Workflow“ anzeigen.



Model Name	Execution Date	Workflow	Approval Status	Remediation Status
LATITUDE 9520	Feb 22, 2025 3:14 AM	View	-	Success
LATITUDE 9520	Feb 21, 2025 12:45 PM	View	-	Success
LATITUDE 9520	Feb 21, 2025 12:18 PM	View	-	Success
PRECISION 3581	Feb 21, 2025 11:29 AM	View	-	Success
PRECISION 3581	Feb 21, 2025 11:14 AM	View	-	Success



Durch das effektive Management dieser Ausgabeoptionen sorgen IT-AdministratorInnen für klare und umsetzbare Ergebnisse in der TechDirect-Benutzeroberfläche, sodass sie ihre PC-Flotte effizient überwachen und korrigieren können.

Erstellen eines Workflows mit dem nutzerdefinierten Workflow-Canvas

Hinweis: Die Korrekturplattform verwendet PowerShell 7, um nutzerdefinierte Skripte auszuführen. Stellen Sie sicher, dass Skripte für PowerShell 7 entwickelt wurden, da Skripte, die für ältere Versionen entwickelt wurden, zu Fehlern oder unerwartetem Verhalten führen können.

Führen Sie die folgenden Schritte aus, um ein nutzerdefiniertes Skript mithilfe des benutzerdefinierten Workflow-Canvas zu laden und auszuführen. Zunächst sehen Sie ein Beispiel für ein einfaches Skript, das ohne untergeordnete Knoten oder erweiterte Verschachtelungslogik ausgeführt wird.

1

PowerShell-Skript hochladen

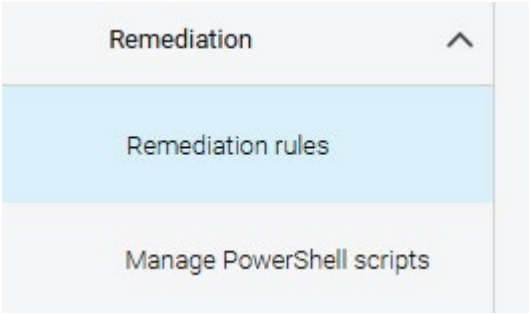
Laden Sie zunächst das PowerShell-Skript mithilfe einer der beiden verfügbaren Methoden auf die Korrekturplattform hoch. Stellen Sie sicher, dass das Skript signiert ist, falls erforderlich, oder lassen Sie es optional unsigniert, wenn dies nicht erforderlich ist, bevor Sie fortfahren.

2

Abschnitt für die Korrektur aufrufen

Navigieren Sie zu TechDirect:

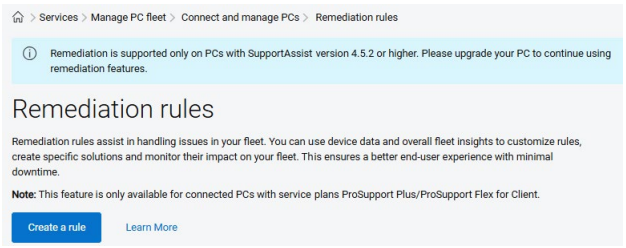
- Wählen Sie den Abschnitt **Remediation** im Dashboard **Connect and manage PCs**.
- Klicken Sie auf **Remediation rules**.



3

Eine neue Regel erstellen

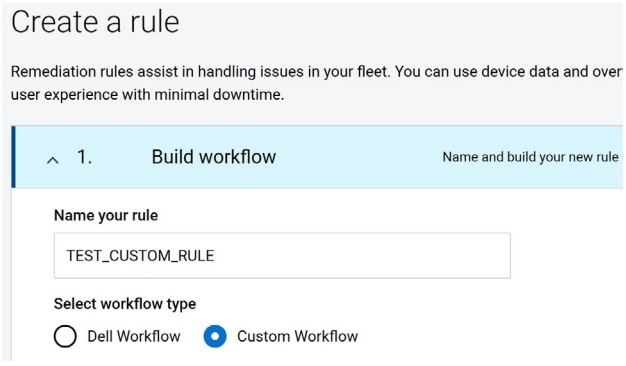
Klicken Sie auf die blaue Schaltfläche **Create a rule**, um den Prozess zum Erstellen einer Regel zu starten.



4

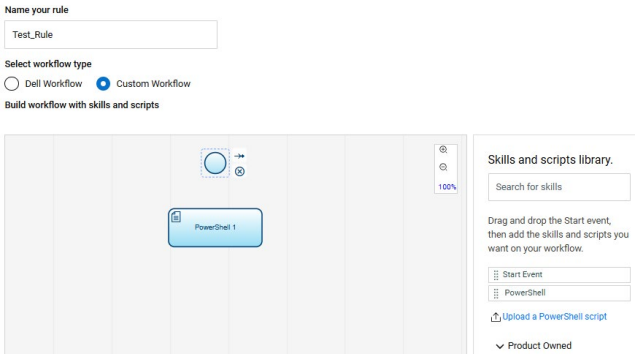
Regel konfigurieren

- Geben Sie einen **Namen** für die Regel ein.
- Wählen Sie **Custom Workflow** als Workflowtyp aus. Dadurch wird das Workflow-Canvas geöffnet.



5 Workflowelemente hinzufügen

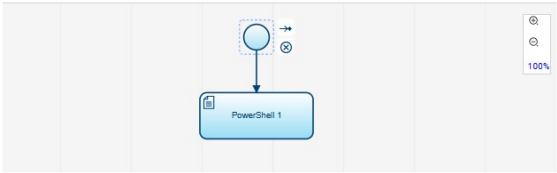
Ziehen Sie für ein einfaches Skript ein Startereignis (ein schattierter Kreis) und ein PowerShell-Konstrukt (ein schattiertes Rechteck mit der Bezeichnung „PowerShell“) aus dem rechten Fensterbereich auf das Canvas.



6 Workflowelemente verknüpfen

Verbinden Sie die Elemente:

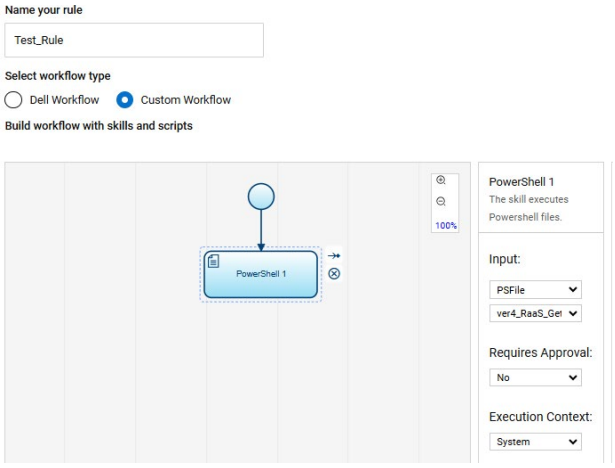
- Klicken Sie auf **Start Event**, ziehen Sie den nach rechts zeigenden Pfeil und verbinden Sie ihn mit dem PowerShell-Konstrukt. Dadurch wird der Workflow eingerichtet.



7 PowerShell-Attribute definieren

Klicken Sie auf das **PowerShell**-Konstrukt und konfigurieren Sie die Attribute im Fensterbereich auf der rechten Seite:

- **Eingabetyp:** Wählen Sie **PSFile (PowerShell)**.
- **Skript:** Wählen Sie das hochgeladene PowerShell-Skript im Dropdown-Menü aus.
- **Genehmigung:**
  - Wählen Sie **Yes**, wenn eine manuelle Genehmigung erforderlich ist, d. h. das Skript wird so lange zurückgestellt, bis es manuell im Abschnitt für Korrekturen freigegeben wird.
  - Wählen Sie **No**, um das Skript automatisch wie geplant oder sofort auszuführen.
- **Ausführungskontext:** Wählen Sie zwischen folgenden Optionen:
  - **System:** Wird mit Administratorrechten ausgeführt (z. B. für den Zugriff auf BitLocker-Schlüssel).
  - **Current User:** Wird unter den Berechtigungen des bzw. der aktiven NutzerIn ausgeführt, was bestimmte Aktionen einschränken kann.



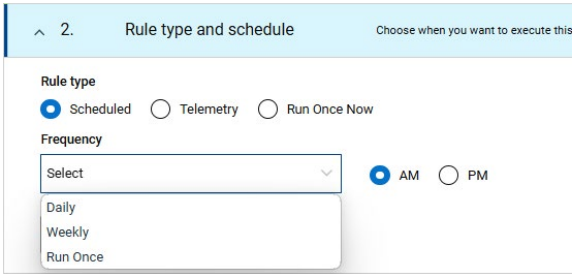
8

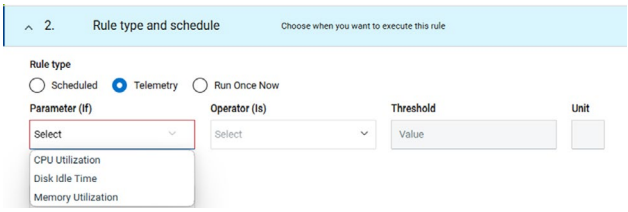
Zeitplan festlegen

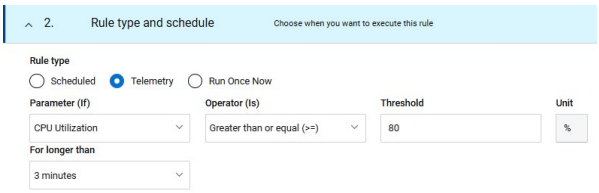
Wählen Sie eine Planungsoption aus:

- **Scheduled:** Definieren Sie, wann das Skript ausgeführt wird:
  - **Run Once:** Wählen Sie ein bestimmtes Datum und eine bestimmte Uhrzeit (AM/PM) aus.
  - **Daily or Weekly:** Definieren Sie ein Zeitfenster (vormittags/nachmittags) und wählen Sie bei wöchentlichen Zeitplänen den Wochentag aus.
  - Beachten Sie, dass Zeitpläne für vormittags zwischen 7:00 und 12:00 Uhr und Zeitpläne für nachmittags zwischen 12:00 und 18:00 Uhr mit zufälligen Startzeiten ausgeführt werden, um Ressourcenkonflikte zu vermeiden.
- **Telemetry-Based Execution:** Löst das Skript automatisch basierend auf Geräteattributen wie den folgenden aus:
  - **CPU Utilization**
  - **Disk Idle Time**
  - **Memory Utilization**
  - Legen Sie einen Schwellenwert fest (z. B. CPU ≥ 80 %) und definieren Sie optional eine Dauer (z. B. länger als drei Minuten).
- **Run Once Now:** Führt das Skript sofort auf Geräten aus, die online sind.

  Um beispielsweise das nutzerdefinierte Skript auszulösen, wenn die CPU-Auslastung eines Geräts länger als drei Minuten 80 % oder mehr beträgt, konfigurieren Sie die Optionen wie folgt:



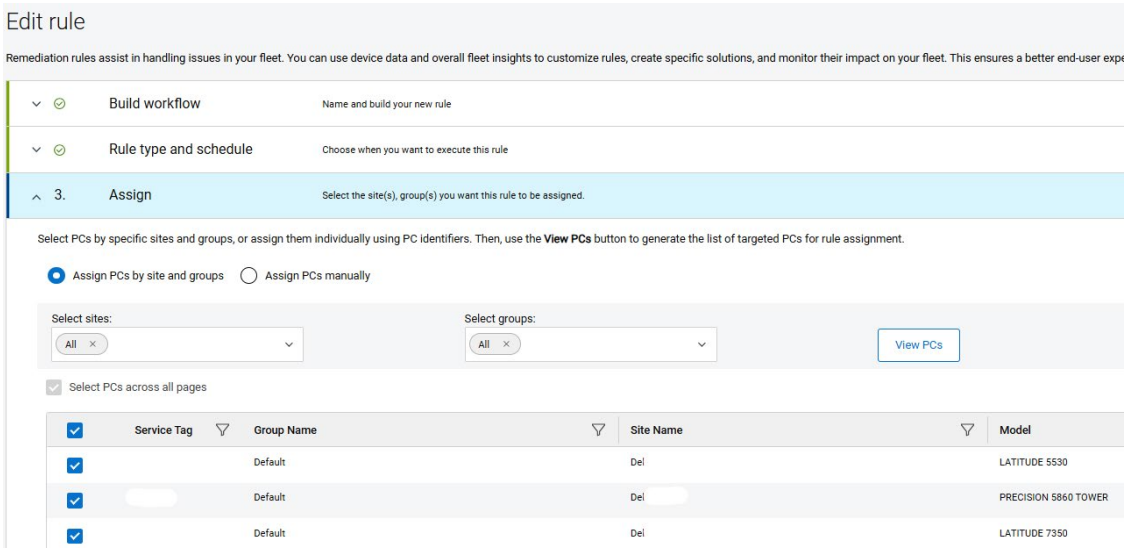


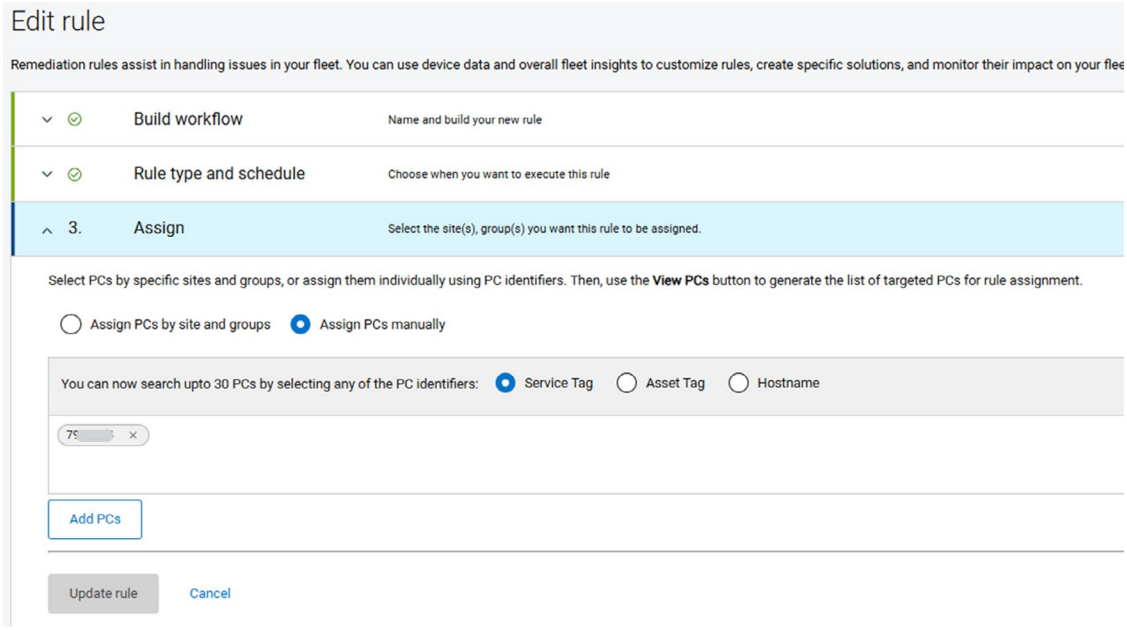


9 Ziel definieren

Wählen Sie die Geräte aus, auf die die Regel angewendet werden soll:

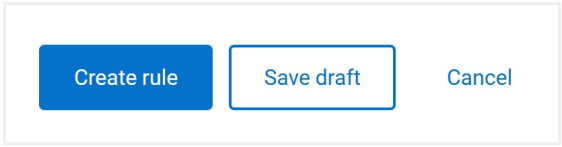
Wählen Sie einen **Standort** oder eine bestimmte **Gruppe** innerhalb eines Standorts aus (z. B. eine Test- oder Produktionsgruppe). Gruppen müssen im Abschnitt zum **Verbinden und Verwalten** vordefiniert sein.





10 Regel finalisieren

Klicken Sie zum Speichern auf **Create Rule**. Wenn die Schaltfläche ausgegraut ist, stellen Sie sicher, dass alle Pflichtfelder (z. B. für Name, Zeitplan, Ziel) ausgefüllt sind.



11 Ergebnisse überwachen

Nach Ausführung der Regel:

- Navigieren Sie zum Abschnitt **Remediation rules** in TechDirect.
- Klicken Sie auf den Regelnamen, um den Status der PCs anzuzeigen:
  - **Der Status „Success", „Failed"** oder **„In Progress"** wird für jeden PC in der Gruppe angezeigt.
- Kehren Sie bei geplanten Regeln zu diesem Abschnitt zurück, um die aktualisierten Ergebnisse nach der Ausführung zu überprüfen.

Mit diesen Schritten können IT-AdministratorInnen nutzerdefinierte Workflows erstellen und managen, um Korrekturen effektiv und fehlerfrei durchzuführen.

Erweitertes übergeordnetes/untergeordnetes PowerShell-Scripting in nutzerdefinierten Korrekturworkflows

Hinweis: Die Korrekturplattform verwendet derzeit PowerShell 7 für die Ausführung nutzerdefinierter Skripte. Stellen Sie sicher, dass alle Skripte in dieser Version entwickelt und getestet wurden, um Kompatibilitätsprobleme zu vermeiden. Die Verwendung älterer Versionen oder der „On-Box"-PowerShell-Umgebung kann zu unerwartetem Verhalten oder Fehlern führen.

Wichtige Terminologie

Für das Verständnis dieses Themas sind mehrere wichtige Begriffe unerlässlich:

- **Übergeordnetes Skript:** Ein Skript der obersten Ebene, das nach seiner Ausführung Exit-Codes sendet oder Zeichenfolgen ausgibt (mithilfe von „Write-Host"). Diese Ausgaben können als Auslösepunkte für untergeordnete Skripte oder vordefinierte Korrekturroutinen dienen.
- **Untergeordnetes Skript:** Ein Skript auf einer niedrigeren Ebene, das als Reaktion auf eine vom übergeordneten Skript festgelegte Bedingung oder einen Auslöser ausgeführt wird.
- **Skill:** Vordefinierte, von Dell bereitgestellte Routinen, die als Alternative zum Erstellen nutzerdefinierter Skripte in Workflows integriert werden können.
- **Verschachtelte Logik:** Eine Struktur, in der über- und untergeordnete Skripte oder Skills mithilfe von bedingter Logik (z. B. WENN/DANN/SONST) interagieren. Dadurch können Workflows dynamisch an bestimmte Bedingungen angepasst werden.

Erstellen einer verschachtelten Logik mit übergeordneten und untergeordneten Skripten

Anpassungsfähige Workflows können durch die Kombination von über- und untergeordneten Skripten mit Korrekturmaßnahmen über eine verschachtelte Logik erstellt werden. Die Ergebnisse oder Ausgaben eines übergeordneten Skripts, z. B. ein Exit-Code oder eine Write-Host-Anweisung, können nachfolgende Skripte oder vordefinierte Korrekturaufgaben auslösen.

Um dies zu veranschaulichen, betrachten Sie einen Beispielworkflow, der zum Managen des Status einer Anwendung erstellt wurde:

1. Der Workflow umfasst ein Startereignis, vier PowerShell-Skripte und einen Korrektur-Skill, die alle per Drag-and-Drop aus dem Abschnitt mit Produktn im Besitz in den rechten Fensterbereich des Workflow-Canvas verschoben wurden.

2. Die Logik innerhalb des Workflows könnte folgenden Schritten folgen:

   - **Schritt 1:** Überprüfen Sie, ob eine bestimmte Anwendung installiert ist, und bestätigen Sie, dass der zugehörige Dienst ausgeführt wird.
   - **Schritt 2:** Wenn die Anwendung installiert ist, der Dienst jedoch nicht ausgeführt wird, starten Sie den Dienst.
   - **Schritt 3:** Wenn die Anwendung nicht installiert ist, rufen Sie eine Korrekturaufgabe auf, um sie zu installieren, und überprüfen Sie anschließend, ob der Dienst nach der Installation ausgeführt wird.
   - **Schritt 4:** Wenn der Dienst nach der Installation der Anwendung ausgeführt wird, beenden Sie den Workflow.
   - **Schritt 5:** Wenn der Dienst weiterhin inaktiv ist, warten Sie eine festgelegte Zeitspanne, bevor Sie erneut versuchen, den Dienst zu starten.

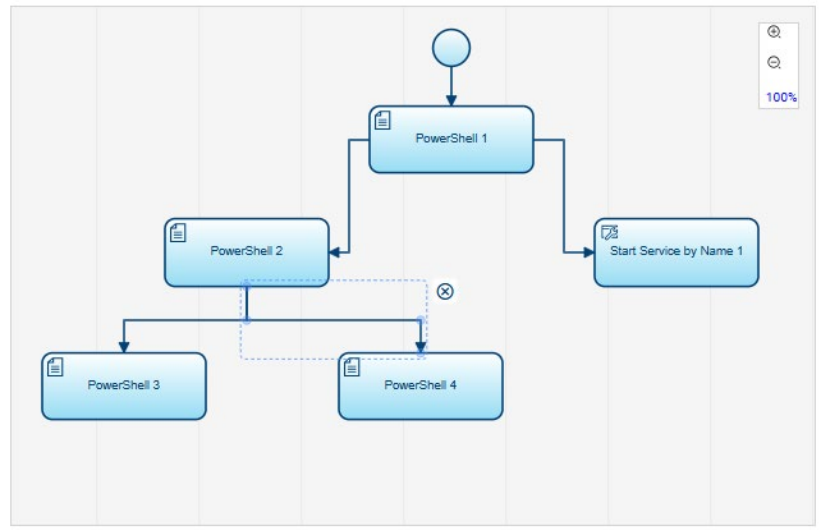
Diese Methodik integriert bedingte Bewertungen, nutzerdefiniertes Scripting und vordefinierte Skills, um Lösungsprozesse zu optimieren. Durch die Nutzung der Flexibilität verschachtelter Logik können Workflows dynamisch auf unterschiedliche Bedingungen reagieren, was zu effizienten und gezielten Korrekturmaßnahmen führt.

Die Nutzung dieser Funktionen ermöglicht die Erstellung robuster Workflows, die auf das Management komplexer Betriebsszenarien zugeschnitten sind und gleichzeitig eine präzise Steuerung durch bedingte Auslöser und strukturierte Logik gewährleisten.

Detaillierte Übersicht über das Canvas-Diagramm und die PowerShell-Workflowimplementierung

Übersicht über das Canvas-Diagramm

Das Canvas-Diagramm ist eine visuelle Darstellung des Workflows mit Verbindungen zwischen verschiedenen Komponenten. In diesem Abschnitt wird der Beispielworkflow wie folgt aufgeschlüsselt:



PowerShell-Workflowschritte

1

PowerShell 1:

- Dieses Skript prüft, ob eine bestimmte Anwendung installiert ist. Wenn die Anwendung installiert ist, wird ermittelt, ob der zugehörige Anwendungsdienst ausgeführt wird.
- **Exit-Status:**
  - Exit 0: Die Anwendung ist installiert und der Dienst wird ausgeführt. Alles funktioniert ordnungsgemäß und der Workflow kann abgeschlossen werden.
  - Exit 1: Die Anwendung ist installiert, aber der Dienst wird nicht ausgeführt.
  - Write-Host-Ausgabe: Sendet die Ausgabe „Anwendung ist nicht installiert“ und löst PowerShell 2 aus.

2

PowerShell 2:

- Wird durch die Write-Host-Ausgabe ausgelöst, die angibt, dass die Anwendung nicht installiert ist. Dieses Skript installiert die Anwendung und sendet nach Abschluss einen der folgenden Exit-Codes:
  - Exit 0: Die Anwendung ist installiert und der Dienst wird ausgeführt.
  - Exit 1: Die Anwendung ist installiert, aber der Dienst wird nicht ausgeführt.

3

PowerShell 3:

- Dieses Skript wird durch Exit 0 nach Abschluss von PowerShell 2 ausgelöst und bestätigt, dass die Anwendung erfolgreich korrigiert wurde. Es sendet eine Meldung und beendet den Workflow.

4

PowerShell 4:

- Dieses Skript wird durch einen Exit 1 nach Abschluss von PowerShell 2 ausgelöst und enthält eine Logik, die zehn Minuten wartet, um der Anwendung bei Bedarf zusätzliche Zeit für die vollständige Initialisierung zu geben. Nach der Wartezeit versucht das Skript, den Anwendungsdienst (z. B. „APP SERVICE NAME“) zu starten. Das Skript wird dann beendet und der Workflow abgeschlossen.

5

Start Service Skill:

- Dieser vordefinierte Skill wird aktiviert, wenn PowerShell 1 den Exit-Code 1 ausgibt, der angibt, dass die Anwendung installiert ist, der zugehörige Dienst jedoch nicht ausgeführt wird. Eingabeparameter wie der Name des Dienstes werden definiert, um den zu startenden Dienst anzugeben. Im Gegensatz zu PowerShell-Skripten stellen Skills vordefinierte Routinen dar, die auf der Korrekturplattform verfügbar sind.

Workflow-Pfadszenarien

1

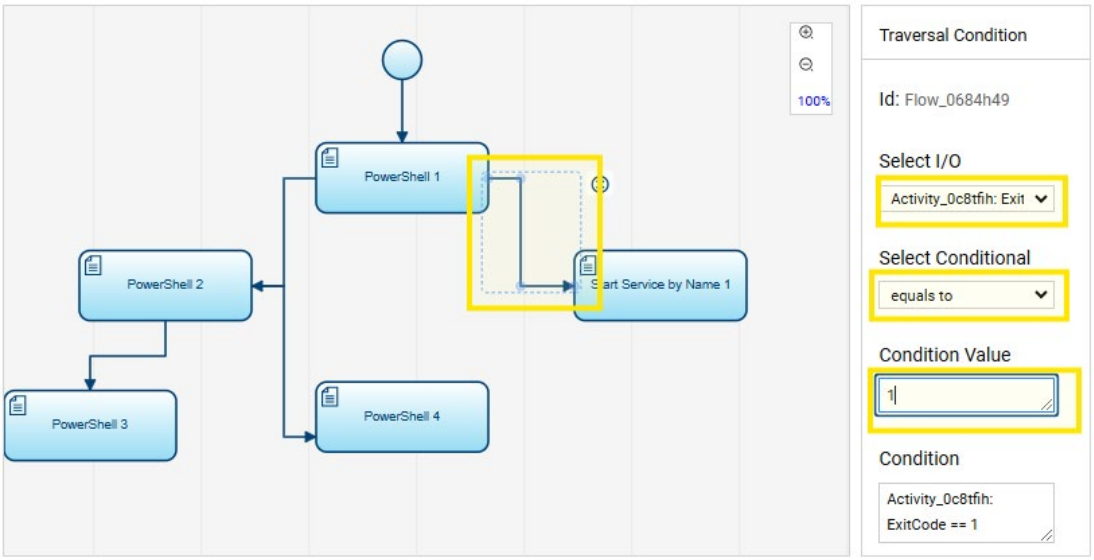
PFAD – PowerShell 1 mit Exit 0:

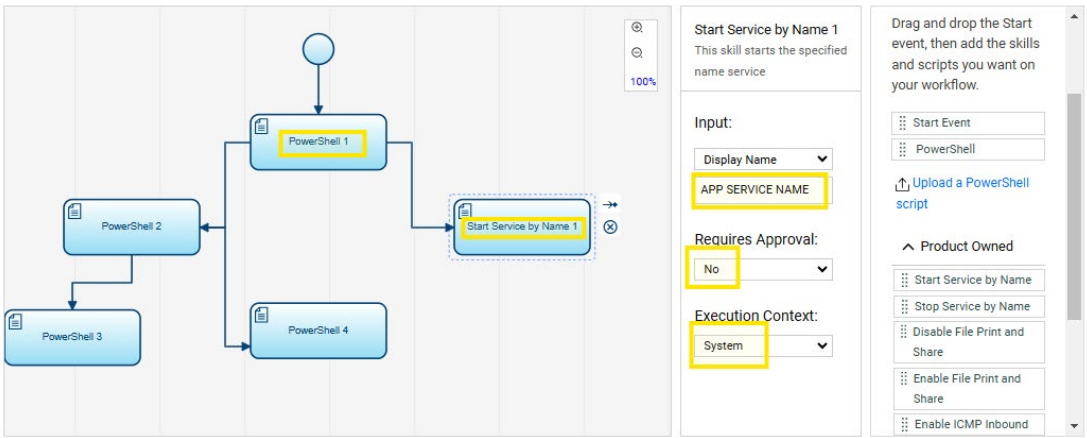
- Wenn PowerShell 1 einen Exit 0 sendet, der angibt, dass die Anwendung installiert ist und der Dienst ausgeführt wird, wird das Skript beendet, ohne weitere Aktionen auszulösen.

2

PFAD – PowerShell 1 mit Exit 1

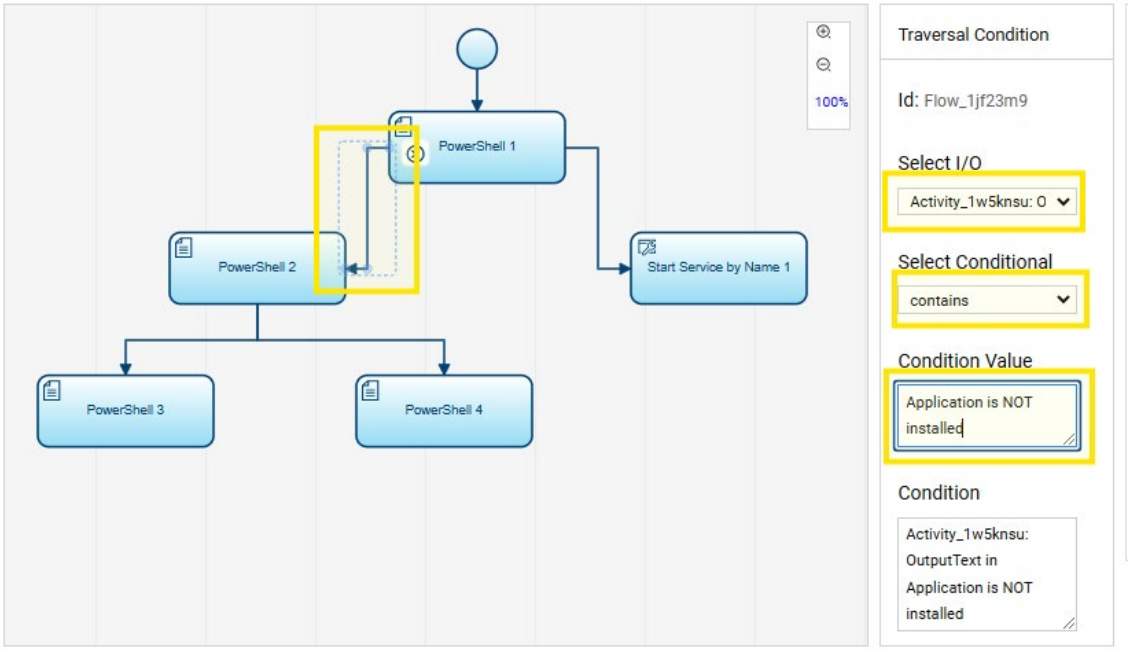
- Wenn Exit 1 erzeugt wird, wechselt der Workflow zum Skill „Dienst starten“. Die Konfiguration des Dienstnamens, der Genehmigung und des Ausführungskontexts für diesen Pfad ist erforderlich, indem die Auslösebedingung im Verbindungspfad definiert wird. Die Schritte umfassen die Auswahl von „Exit“ in der Dropdown-Liste „E/A“ und die Sicherstellung, dass der Exit-Wert gleich 1 ist, was bedeutet, dass die Anwendung installiert ist, der Dienst jedoch nicht ausgeführt wird. Nach der Konfiguration wird der Dienst (z. B. „APP SERVICE NAME“) gestartet und der Workflow abgeschlossen, da keine zusätzliche verschachtelte Logik erforderlich ist.

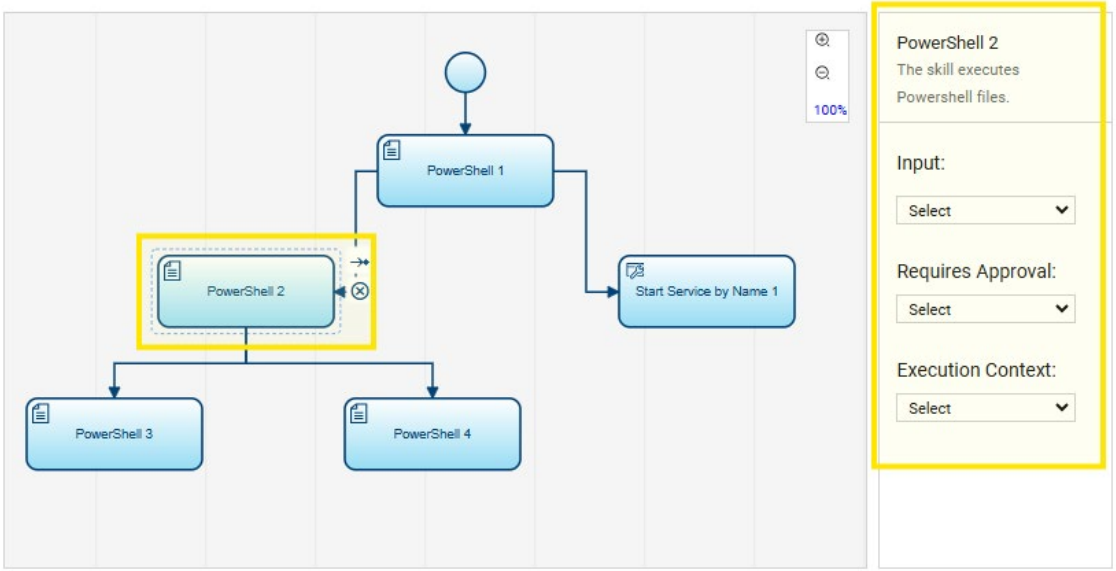




3 PFAD – PowerShell 1 mit Write-Host-Ausgabe

Wenn die Write-Host-Ausgabe „Application is not installed“ anzeigt, wird PowerShell 2 ausgelöst. Die übergeordnete Bedingung wird im Verbindungspfad definiert, indem Sie in der Dropdown-Liste „Output Text“ auswählen, die Bedingung auf „Contains“ setzen und den Wert „Application is not installed“ angeben. Anschließend werden die PowerShell 2-Parameter konfiguriert, um mit der Installation der Anwendung fortzufahren.

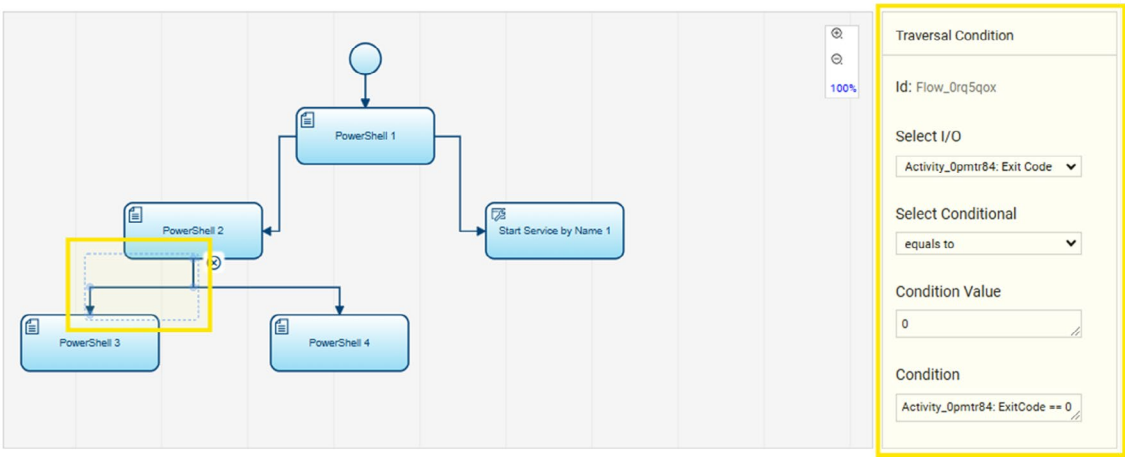


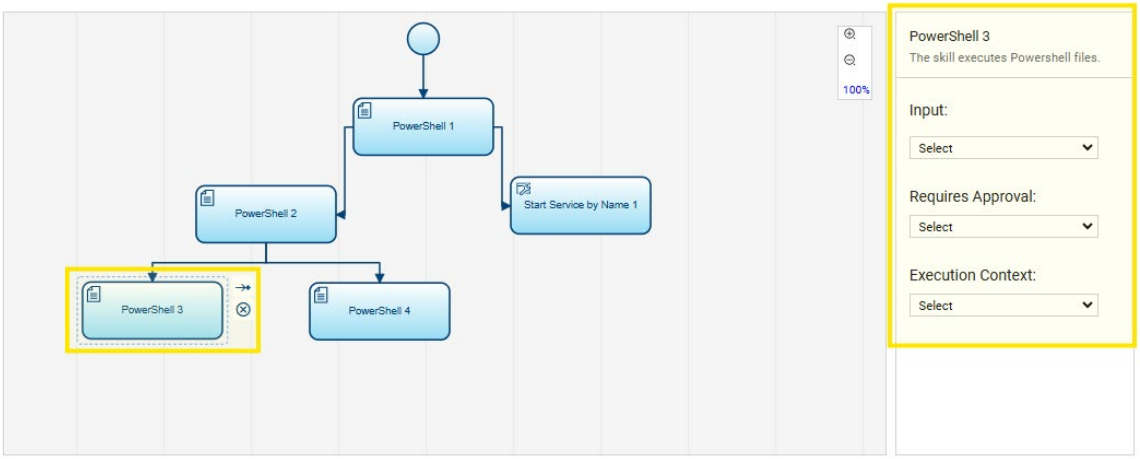


4

PFAD – PowerShell 3 mit Exit 0:

Ein Exit 0 aus PowerShell 2 leitet den Workflow an PowerShell 3 weiter. Der Pfad wird konfiguriert, indem der E/A-Typ als Exit-Code mit einem bedingten Wert von 0 angegeben wird. PowerShell 3 beendet dann den Workflow und zeigt eine Meldung an, die bestätigt, dass die Anwendung repariert wurde und der Dienst ausgeführt wird.

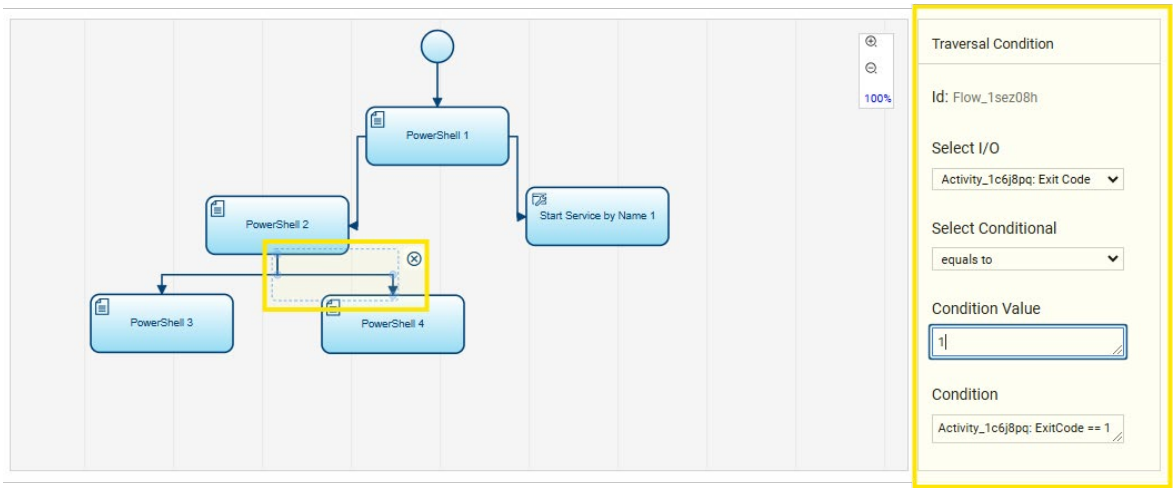


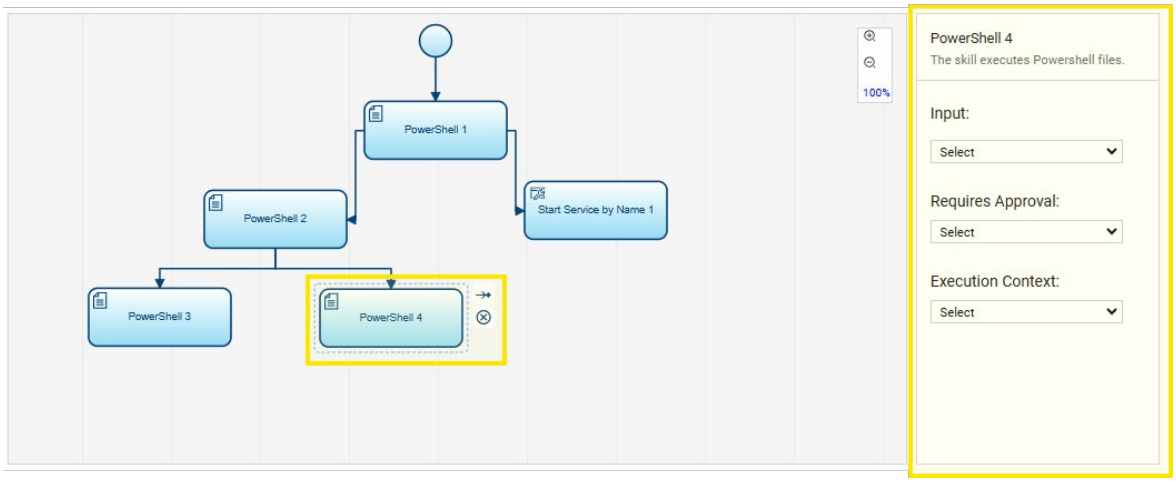


5

PFAD – PowerShell 4 mit Exit 1:

- Ein Exit 1 aus PowerShell 2 leitet den Workflow an PowerShell 4 weiter. Wie andere Pfade wird auch dieser konfiguriert, indem der E/A-Typ „Exit Code“ ausgewählt, die Bedingung auf „Equals“ gesetzt und der Wert 1 zugewiesen wird. Die Logik von PowerShell 4 umfasst eine Wartezeit von zehn Minuten, nach der der Anwendungsdienst gestartet wird und der Workflow beendet wird.





Weitere Überlegungen

1. **Updates der von Dell erstellten Bibliothek**
   Überprüfen Sie regelmäßig die von Dell erstellte Skriptbibliothek auf neu hinzugefügte Korrekturen, die das verfügbare Angebot erweitern.

2. **Ausführungskontexte**
   Die meisten Skripte werden in einem Systemkontext ausgeführt, der den Zugriff auf Informationen auf Administratorebene ermöglicht, die für Endbenutzer nicht verfügbar sind. Bestimmte Skripte werden jedoch in einem Nutzerkontext ausgeführt, um nutzerspezifische Daten zu erfassen, oder unter einem hybriden Ansatz, bei dem verschiedene Teile des Skripts entweder im System- oder Nutzerkontext ausgeführt werden. Wenn ein Nutzerkontext erforderlich ist, muss ein/e angemeldete/r NutzerIn vorhanden sein. Andernfalls gibt das Skript eine Meldung zurück, dass kein/e NutzerIn angemeldet ist.

3. **Firewall- und NGAV-Kompatibilität**
   Stellen Sie sicher, dass SupportAssist zu allen Firewall- oder NGAV-Positivlisten (Next-Gen AV) hinzugefügt wird, falls solche Konfigurationen erforderlich sind.

4. **Skripte zur Datenerfassung**
   Einige Skripte verwenden untergeordnete Skripte zur Datenerfassung, die möglicherweise mit unterschiedlichen Häufigkeiten ausgeführt werden müssen, um umfassendere historische Trends zu liefern. Vermeiden Sie das Entfernen von Dell Korrekturskripten, die als Aufgabenfolgen auf Geräten geplant sind, da diese für die Erfassung und Analyse historischer Daten durch übergeordnete Skripte, die täglich oder wöchentlich geplant sind, unerlässlich sind.

Fazit

SupportAssist-Korrekturskripte bieten IT-AdministratorInnen die Tools, um PC-Flotten effektiv und zuverlässig zu managen und zu optimieren. Durch die Kombination automatisierter Workflows, robuster Sicherheitsprotokolle und detaillierter Transparenz über Protokolle vereinfacht die Plattform alltägliche IT-Prozesse und stellt gleichzeitig sicher, dass die Systeme zuverlässig und sicher bleiben. Ganz gleich, ob Sie von Dell erstellte oder benutzerdefinierte Skripte verwenden – diese leistungsstarke Lösung ermöglicht IT-AdministratorInnen, Herausforderungen effizient zu bewältigen und sich souverän auf strategische Prioritäten zu konzentrieren.

Informationen zu unterstützten Systemen und Anforderungen finden Sie im Benutzerhandbuch (SupportAssist for Home PCs für die persönliche Verwendung) oder im Administratorhandbuch (SupportAssist for Business PCs für ein flottenweites PC-Management) unter „Unterstützte PCs“. Proaktive und vorausschauende Funktionen hängen vom aktiven Serviceplan und den Geschäftsregeln von Dell Technologies ab. Informationen zu den Funktionen der ProSupport Suite for PCs finden Sie im Administratorhandbuch.“ Wählen Sie die Registerkarte „Funktionen für das Verbinden und Verwalten und Dell Servicepläne“ aus“. Um sich über die Funktionen der Dell Care Suite, Premium Support Suite oder Alienware Care Suite for PCs zu informieren, konsultieren Sie das Benutzerhandbuch und klicken Sie auf „SupportAssist-Funktionen und Dell Servicepläne“.