DRUCKSYSTEME
Janz & Raschke GmbH

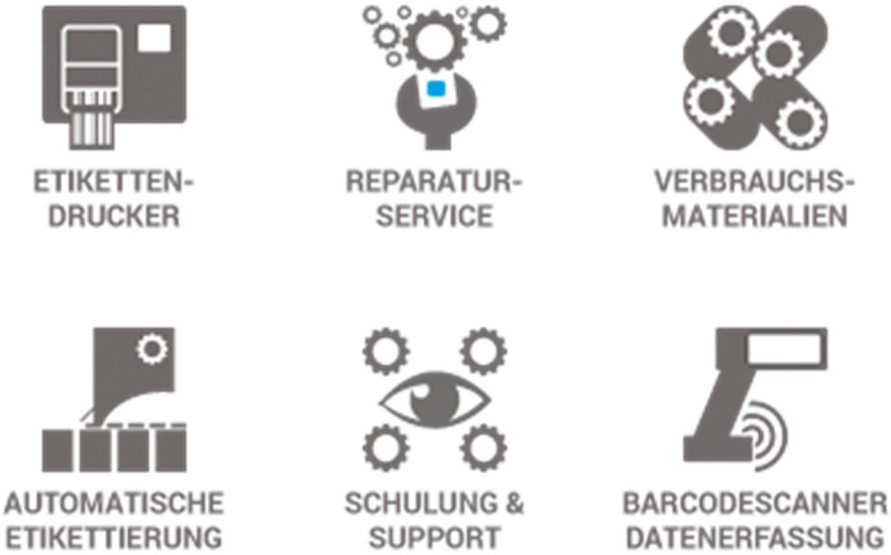
Bedienungsanleitung / Handbuch / Datenblatt

Sie benötigen einen Reparaturservice für Ihren Etikettendrucker

oder suchen eine leicht zu bedienende Etikettensoftware?

Wir helfen Ihnen gerne weiter.

Ihr Partner für industrielle Kennzeichnungslösungen



Drucksysteme Janz & Raschke GmbH
Röntgenstraße 1
D-22335 Hamburg
Telefon +49(0)40 – 840 509 0
Telefax +49(0)40 – 840 509 29

kontakt@jrdrucksysteme.de
www.jrdrucksysteme.de

DRUCKSYSTEME
Janz & Raschke GmbH

Bedienungsanleitung / Handbuch / Datenblatt

Maßgeschneiderte Lösungen für den Etikettendruck und die Warenkennzeichnung

Seit unserer Gründung im Jahr 1997, sind wir erfolgreich als Partner namhafter Hersteller und als Systemintegrator im Bereich der industriellen Kennzeichnung tätig.

Unser Motto lautet:

So flexibel wie möglich und so maßgeschneidert wie nötig.

Ich stehe mit meinem Namen für eine persönliche und kompetente Beratung. Wir hören Ihnen zu und stellen mit Ihnen eine Lösung zusammen, die Ihren individuellen Anforderungen entspricht. Für Sie entwickeln unsere erfahrenen Techniker und Ingenieure neben Etikettiermaschinen, maßgeschneiderte Komplettlösungen inklusive Produkthandling, Automatisierungstechnik und Softwarelösung mit Anbindung an Ihr Warenwirtschaftssystem.

Ich freue mich von Ihnen zu hören.

Bis dahin grüßt Sie

Jörn Janz

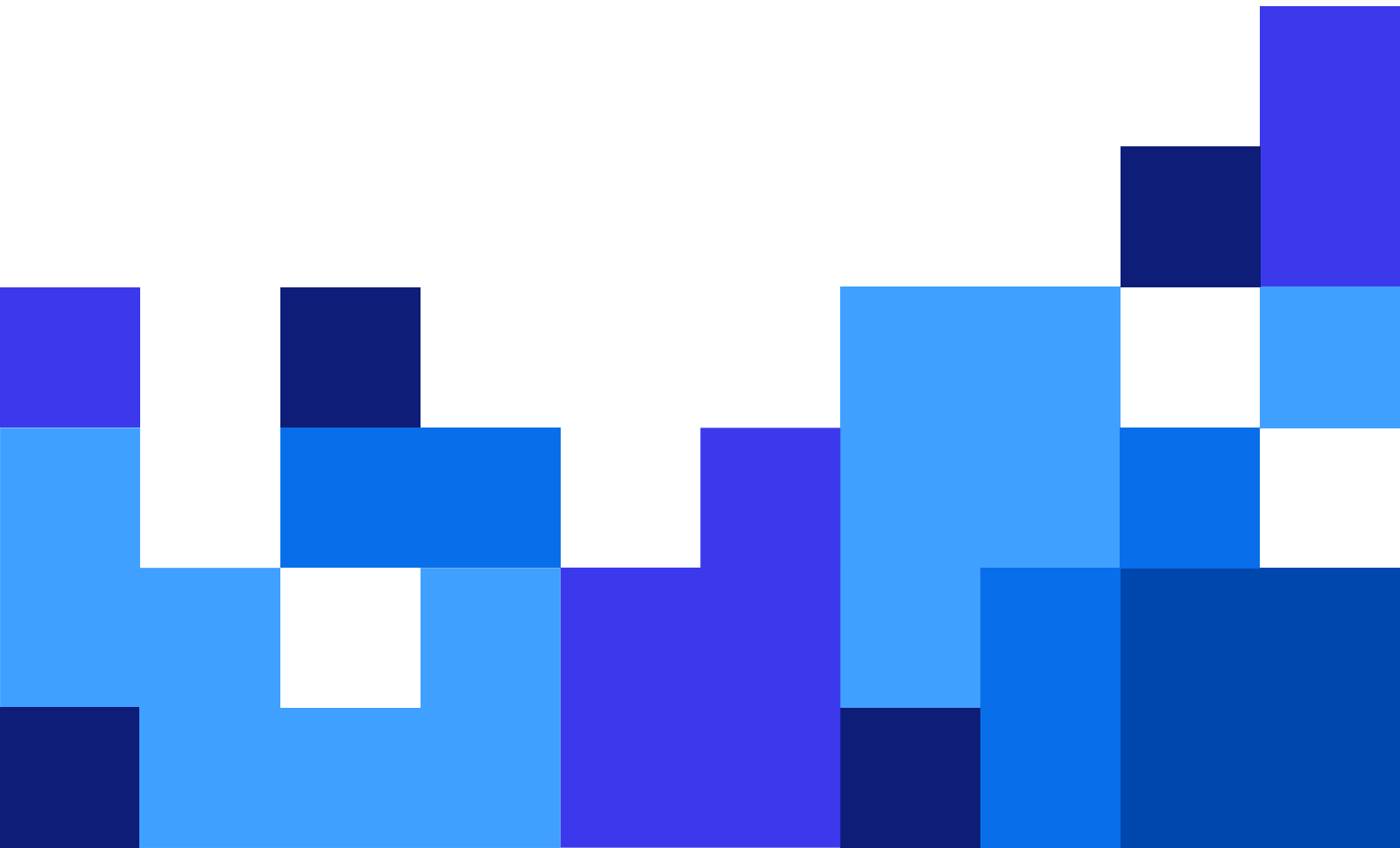
Hier finden Sie Ihren Ansprechpartner:

http://www.jrdrucksysteme.de/kontakt/

Loftware NiceLabel

10 Automation NiceLabel Benutzerhandbuch

Rev-2021-10



Inhaltsverzeichnis

1. Willkommen bei NiceLabel Automation

NiceLabel Automation ist eine Anwendung, die sich wiederholende Aufgaben automatisiert. In den meisten Fällen werden Sie sie verwenden, um Etikettendruckprozesse in vorhandene Informationssysteme zu integrieren, etwa in Geschäftsanwendungen, Produktions- und Verpackungslinien, Verteilungssysteme und Lieferketten. Mit NiceLabel Automation können alle Anwendungen in allen Abteilungen und Niederlassungen Ihres Unternehmens Etiketten mithilfe von autorisierten Etikettenvorlagen drucken.

NiceLabel Automation hilft Ihnen dabei, ein optimales Etikettendruck auf Unternehmensebene bereitzustellen und aufzuführen, indem es Geschäftsereignisse mit der Etikettenproduktion synchronisiert. Automatisierter Druck ohne menschlichen Eingriff ist die mit Abstand effektivste Methode, um Benutzerfehler zu vermeiden und eine maximale Performance zu gewährleisten.

Die Automatisierung des Etikettendrucks anhand einer Trigger-basierten Anwendung dreht sich um die folgenden drei Kernprozesse:

Trigger
Trigger sind eine einfache, aber leistungsstarke Funktion, die Ihnen bei der Automatisierung Ihrer Aufgaben hilft. Grundsätzlich handelt es sich bei einem Trigger um eine Ursache und Wirkung folgender Aussage: „Wenn ein überwachtes Ereignis eintritt, tu etwas bestimmtes."

Hier geht es also um eine **WENN ... DANN**-Verarbeitung. Trigger sind optimal auf die Handhabung wiederkehrender Ereignisse ausgelegt.

Der automatisierte Etikettendruck wird durch ein bestimmtes Geschäftsereignis ausgelöst. NiceLabel Automation wird dafür konfiguriert, einen Ordner, eine Datei oder eine Kommunikationsschnittstelle zu überwachen. Tritt ein bestimmtes Ereignis ist, werden eine Änderung an einer Datei oder eingehende Daten verzeichnet. Dies löst den Etikettendruckprozess aus.

Im entsprechenden Kapitel finden Sie weitere Informationen zu den verschiedenen Arten von Triggern:

- Dateitrigger
- Trigger für serielle Schnittstelle
- Datenbank-Trigger
- Planer-Trigger
- TCP/IP Server Trigger
- TCP/IP Client Trigger
- HTTP-Trigger
- Webdienst-Trigger
- Cloud-Trigger

Datenextraktion und -Platzierung
Sobald ein Trigger den Druck einleitet, extrahiert NiceLabel Automation Etikettendaten und fügt sie in variable Objekte ein, die auf einem Etikett platziert werden.

Filter für die Datenextraktion unterstützen:

- Strukturierte Textdateien
- Unstrukturierte Textdateien
- Verschiedene XML-Dateien
- JSON-Dateien

Ausführen von Aktionen
Nach Abgleich der Daten mit variablen Objekten auf einem Etikett, beginnt NiceLabel Automation mit der Ausführung von Aktionen. Einfache Aktionsfolgen beinhalten meistens die Aktionen **Etikett öffnen** und **Etikett drucken**, um die extrahierten Daten auf ein Etikett zu drucken. Außerdem können Sie Daten an benutzerdefinierte lokale oder Netzwerk-Speicherorte, Webserver, Hardwaregeräte und vieles mehr senden.

Es stehen über 30 Aktionen zur Verfügung. Gemeinsam decken Sie einen Großteil der Szenarien ab, die für heutige Geschäftsumgebungen relevant sind.

Weitere Informationen über grundlegende und erweiterte Aktionen.

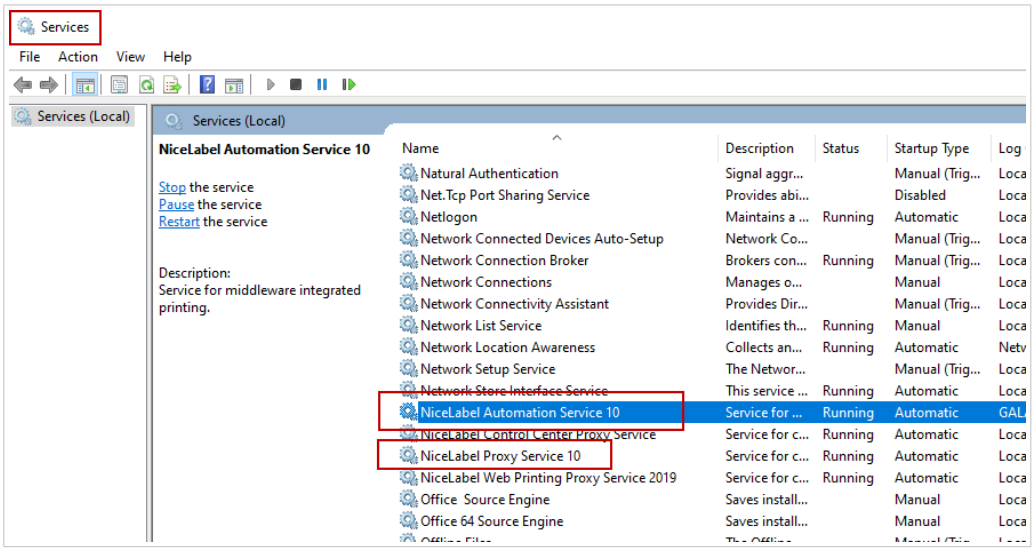
1.1. Architektur

NiceLabel Automation ist eine dienstbasierte Anwendung. Die Ausführung aller Regeln und Aktionen erfolgt als Hintergrundprozess anhand der Zugangsdaten des Benutzerkontos, das für den Dienst festgelegt ist.

NiceLabel Automation besteht aus drei Komponenten.

- Automation Builder**:** Entwickler nutzen diese Anwendung, um Trigger, Filter und Aktionen zu erstellen und in einer funktionierenden Konfiguration einzusetzen. Aktionen, die zu einer solchen Konfiguration gehören, werden ausgeführt, nachdem ein Trigger Daten empfangen hat. Diese Anwendung wird immer im 32-Bit-Modus ausgeführt.
- Automation Manager**:** Dies ist die Verwaltungsanwendung, die die Ausführung von Triggern in Echtzeit überwacht und die Trigger anweist, zu starten oder anzuhalten. Automation Manager wird immer als 32-Bit-Anwendung ausgeführt.
- NiceLabel Automation **-Dienst:** Dies ist die „Druck-Engine“, die die in den Triggern definierten Regeln ausführt. Es gibt zwei Dienstanwendungen: NiceLabel Automation Service und NiceLabel Proxy Service. Service erkennt den Bit-Modus des Windows-Rechners und wird entsprechend

ausgeführt (z. B. als 64-Bit-Anwendung unter 64-Bit-Windows), während Proxy Service immer als 32-Bit-Prozess ausgeführt wird.



1.2. Systemanforderungen

ANMERKUNG

Prüfen Sie die aktuellen Systemanforderungen auch immer auf dieser Webseite: Systemanforderungen.

1.3. Installation

ANMERKUNG

Im Folgenden finden Sie die zusammengefasste Version des Installationsverfahrens. Weitere Informationen finden Sie im NiceLabel Automation Installationshandbuch.

Stellen Sie vor Beginn der Installation sicher, dass Ihre Infrastruktur mit kompatibel ist Systemanforderungen.

So installieren Sie NiceLabel Automation:

- Laden Sie die Installationsdatei auf der Website NiceLabel herunter, und starten Sie danach die heruntergeladene ausführbare Datei.
- Legen Sie die NiceLabel-DVD ein.

    1. Das Anwendungs-Hauptmenü startet automatisch.
    Ist dies nicht der Fall, doppelklicken Sie auf die Datei **START . EXE** auf der DVD.

2. Klicken Sie auf NiceLabel **Installieren**.
3. Folgen Sie den Anweisungen des **Assistenten**.
   Während der Installation fordert der Assistent Sie zur Eingabe des Benutzernamens auf, unter dem der NiceLabel Automation-Dienst ausgeführt werden soll. Wählen Sie auf jeden Fall einen echten Benutzernamen ein, da der Dienst die mit dem jeweiligen Benutzernamen verbundenen Berechtigungen erhält. Weitere Informationen finden Sie im Abschnitt Im Dienstmodus ausführen.

Versions-Upgrade

Um ein Upgrade von NiceLabel Automation auf die neue Version durchzuführen, installieren Sie die neue Version einfach über die installierte Version, um diese zu überschreiben. Während des Upgrades wird die alte Version entfernt und durch die neue ersetzt. Alle vorhandenen Einstellungen werden beibehalten. Beim Upgrade wird der Inhalt der Protokolldatenbank gelöscht.

1.4. Aktivierung

Aktivieren Sie NiceLabel Automation, um die Verarbeitung konfigurierter Trigger zu aktivieren. Der Aktivierungsvorgang erfordert eine Internetverbindung – nach Möglichkeit auf demselben Rechner, auf dem die Software ausgeführt wird. Die Aktivierung der Testlizenz erfolgt anhand desselben Verfahrens.

ANMERKUNG

Sie können die Software entweder aus Automation Builder oder aus Automation Manager aktivieren.

Aktivierung in Automation Builder

1. Starten Sie Automation Builder.
2. Wählen Sie **Datei>Über>Ihre Lizenz aktivieren**.
   Dadurch wird der Aktivierungs-Assistent gestartet.
3. Folgen Sie den Anweisungen auf dem Bildschirm.

Aktivierung in Automation Manager

1. Starten Sie Automation Manager.
2. Gehen Sie zur Registerkarte **Über**.
3. Klicken Sie auf **Ihre Lizenz aktivieren**.
4. Folgen Sie den Anweisungen auf dem Bildschirm.

1.5. Testmodus

Im Testmodus können Sie NiceLabel Automation bis zu 30 Tage lang testen. Im Testmodus stehen Ihnen dieselben Funktionen zur Verfügung wie in einer lizenzierten Version, damit Sie das Produkt vor dem Kauf umfassend testen können. Automation Manager Zeigt durchgehend einen Hinweis auf die Testversion und die Anzahl von verbleibenden Tagen an. Nach Ablauf des Testzeitraums verarbeitet der NiceLabel Automation-Dienst keine Trigger mehr. Der 30-Tage-Zeitraum beginnt am Tag der Installation.

ANMERKUNG

Sie können den Testzeitraum verlängern, indem Sie Ihren NiceLabel Händler kontaktieren und einen zusätzlichen Testlizenzschlüssel anfordern. Sie müssen den Testlizenzschlüssel aktivieren. Weitere Informationen finden Sie im Abschnitt Aktivierung.

1.6. Datei-Tab

Das **Datei**-Tab dient als Bereich zur Verwaltung von Dokumenten. Die folgenden Optionen sind verfügbar:

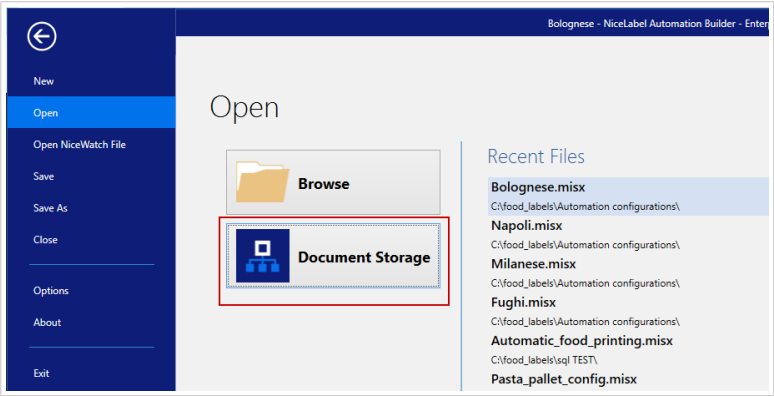
- **Neu**: erstellt eine neue Konfigurationsdatei.
- Öffnen: öffnet vorhandene Konfigurationsdateien.
- **NiceWatch Datei öffnen**: öffnet eine alte NiceLabel NiceWatch-Konfiguration.
- Speichern: speichert die aktive Konfigurationsdatei.
- Speichern unter: ermöglicht es Ihnen, die aktive Konfigurationsdatei unter Angabe eines Namens und Speicherorts zu speichern.
- Optionen: öffnet den Dialog, in dem die Standardeinstellungen für das Programm vorgenommen werden.
- Über: bietet Informationen zur Lizenz und Softwareversion.
- **Beenden**: schließt die Anwendung.

1.6.1. Öffnen

Mit dem **Öffnen-Dialog** können Sie die vorhandenen Konfigurationen in Automation Builder öffnen.

Durchsuchen ermöglicht die Auswahl der Konfigurationsdateien auf lokalen oder Netzwerk-Festplatten.

Dokumentenspeicher öffnet den Speicherort des Dokumentenspeichers im verbundenen NiceLabel Control Center. Die Registerkarte **Dokumentenspeicher** ermöglicht es Ihnen, Ihre Kopie der gespeicherten Konfigurationsdatei zu verwalten.



Im Feld **Letzte Dateien** sind die Konfigurationsdateien aufgelistet, die zuletzt bearbeitet wurden. Klicken Sie auf eine von ihnen, um die entsprechende Datei zu öffnen.

1.6.2. Kompatibilität mit NiceWatch-Produkten

NiceLabel Automation Ermöglicht es Ihnen, Trigger-Konfigurationen zu laden, die mit älteren NiceWatch Produkten erstellt wurden. In den meisten Fällen können Sie die NiceWatch-Konfiguration anhand von NiceLabel Automation ausführen, ohne Anpassungen vorzunehmen.

NiceLabel Automation-Produkte nutzen eine neue .NET-basierte Druck-Engine, die für Performance und geringe Arbeitsspeicherbelastung optimiert wurde. Die neue Druck-Engine unterstützt nicht alle Etikettendesign-Optionen, die im Etiketten-Designer zur Verfügung stehen. Diese Lücken werden in jeder neuen Version von NiceLabel Automation nach und nach gefüllt, aber es gibt möglicherweise immer noch einige nicht verfügbare Funktionen.

Kompatibilitätsprobleme beheben

NiceLabel Automation warnt Sie, wenn Sie versuchen, vorhandene Etikettenvorlagen zu drucken, die Funktionen beinhalten, welche von der neuen Druck-Engine nicht unterstützt werden.

Automation benachrichtigt Sie über folgende Inkompatibilitäten mit der NiceWatch-Konfiguration oder mit Etikettenvorlagen:

- **Kompatibilität mit der Trigger-Konfiguration:** Wenn Sie die NiceWatch-Konfiguration (.MIS file) öffnen, gleicht NiceLabel Automation sie mit den unterstützten Funktionen ab. Nicht alle Funktionen von NiceWatch stehen in NiceLabel Automation zur Verfügung. Einige sind überhaupt nicht verfügbar, während andere nur unterschiedlich konfiguriert werden. Wenn die MIS-Daten nicht unterstützte Funktionen enthält, werden diese aufgelistet. Automation entfernt diese Funktionen aus der Konfiguration.
  In solchen Fällen müssen Sie die .MIS-Datei in Automation Builder öffnen und die Kompatibilitätsprobleme beheben. Sie müssen die verfügbaren Funktionen von NiceLabel Automation nutzen, um die entfernten Konfigurationselemente neu zu erstellen.
- **Kompatibilität mit Etikettenvorlagen:** Wenn Ihre vorhandenen Etikettenvorlagen Druck-Engine-Funktionen enthalten, die von NiceLabel Automation nicht unterstützt werden, werden Fehlermeldungen im **Log**-Bereich angezeigt. Diese Information wird im Automation Builder (beim Erstellen von Triggern) oder in Automation Manager (beim Ausführen von Triggern) angezeigt.
  In diesem Fall müssen Sie die Etikettendatei im Etiketten-Designer öffnen und die nicht unterstützten Funktionen aus dem Etikett entfernen.

ANMERKUNG

Weitere Informationen über Kompatibilitätsprobleme mit NiceWatch und Etiketten-Designern finden Sie im NiceLabel Automation Compatibility with NiceLabel Designers V6 and NiceWatch V5.

NiceWatch-Konfiguration zur Bearbeitung öffnen

Öffnen Sie die vorhandene NiceWatch-Konfiguration (.MIS Datei) in Automation Builder und bearbeiten Sie sie in Automation Builder. Sie können die Konfiguration nur als MISX-Datei speichern.

So bearbeiten Sie die NiceWatch-Konfiguration:

1. Starten Sie Automation Builder.
2. Wählen Sie **Datei** > **NiceWatch Datei öffnen**.
3. Wählen Sie im Dialogfeld **Öffnen** die gewünschte NiceWatch-Konfigurationsdatei (.MIS Datei) aus.
4. Klicken Sie auf **OK**.
5. Wenn die Konfiguration nicht unterstützte Funktionen beinhaltet, werden sie in einer Liste angezeigt. Automation entfernt sie aus der Konfiguration.

NiceWatch-Konfiguration zur Ausführung öffnen
Sie können die NiceWatch-Konfiguration (.MIS-Datei) in Automation Manager öffnen, ohne sie ins NiceLabel Automation Dateiformat (.MISX-Datei) zu konvertieren. Wenn die Trigger aus NiceWatch mit NiceLabel Automation kompatibel sind, können Sie sie auf Anhieb verwenden.

Um die NiceWatch-Konfiguration zu öffnen und zu implementieren, tun Sie Folgendes:

1. Starten Sie Automation Manager.
2. Klicken Sie auf die **Hinzufügen (+)**-Schaltfläche.
3. Ändern Sie im Dialogfeld **Öffnen** den Dateityp zu **NiceWatch Konfiguration**.
4. Navigieren Sie zur NiceWatch-Konfigurationsdatei (.MIS-Datei).
5. Klicken Sie auf **OK**.
6. Automation Manager zeigt den Trigger für die ausgewählte Konfiguration an. Um den Trigger zu starten, wählen Sie ihn aus und klicken Sie auf **Start**.

ANMERKUNG
Falls Kompatibilitätsprobleme mit der NiceWatch-Konfiguration bestehen, müssen Sie sie in Automation Builder öffnen und umkonfigurieren.

1.6.3. Speichern

Speichern speichert die aktive Konfigurationsdatei unter demselben Dateinamen, unter dem sie geöffnet wurde.

ANMERKUNG
Wird eine Konfiguration zum ersten Mal geöffnet, gelangen Sie über die **Speichern**-Option zum Dialog **Speichern unter**.

1.6.4. Speichern unter

Speichern unter ermöglicht es Ihnen, die aktive Konfigurationsdatei unter Angabe eines Namens und Speicherorts zu speichern. Ihre Konfigurationsdateien können auf Ihrem lokalen Datenträger oder im **Dokumentenspeicher** Ihres NiceLabel Control Center gespeichert werden.

Im Feld **Letzte Ordner** sind die Ordner aufgelistet, die vor Kurzem zum Speichern von Konfigurationsdateien verwendet wurden.

1.6.5. Optionen

Nutzen Sie die Einstellungen in diesem Dialogfeld, um die Anwendung anzupassen. Wählen Sie eine Gruppe aus dem linken Bereich und konfigurieren Sie ihre Einstellungen.

Ordner

In diesem Bereich können Sie die Standard-Speicherordner für Etiketten, Masken, Datenbanken und Bilddateien auswählen. Der Standardspeicherort ist der Dokumente-Ordner des aktuellen Benutzers. Es gibt außerdem Standardordner, in denen NiceLabel Automation nach Dateien sucht, wenn Sie Dateinamen ohne vollständigen Pfad angeben. Weitere Informationen über die Reihenfolge bei der Dateisuche finden Sie in Abschnitt Suchreihenfolge für die angeforderten Dateien.

Ordner-bezogene Änderungen werden innerhalb von einer Minute in NiceLabel Automation übernommen. Um Änderungen umgehend anzuwenden, starten Sie den Dienst neu.

ANMERKUNG

Die Einstellungen, die Sie hier anwenden, werden im Profil des momentan angemeldeten Benutzers gespeichert. Falls Ihr NiceLabel Automation-Dienst unter einem anderen Benutzeraccount ausgeführt wird, müssen Sie sich mit diesem anderen Account bei Windows anmelden, bevor Sie den Standard-Etikettenordner ändern können. Sie können auch das Windows Befehlszeilen-Dienstprogramm RUNAS verwenden, um Automation Builder als dieser andere Benutzer auszuführen.

Sprache

Im Sprache-Bereich können Sie die Sprache für die Benutzeroberfläche von NiceLabel Automation auswählen. Wählen Sie die gewünschte Sprache aus und klicken Sie auf **OK**.

ANMERKUNG

Die Änderung wird wirksam, wenn Sie die Anwendung neu starten.

Globale Variablen

Der Bereich „Globale Variablen“ legt fest, welcher Speicherort für globale Variablen verwendet werden soll:

- **Globale Variablen verwenden, die auf dem Server gespeichert sind (NiceLabel Control Center):** Stellt den Speicherort für globale Variablen auf den NiceLabel Control Center Server ein.

  PRODUKTEBENEN-INFO

  Diese Option ist mit LMS Pro oder NiceLabel LMS Enterprise Lizenzen verfügbar.

- **Globale Variablen verwenden, die in einer Datei gespeichert sind (lokal oder in Freigabe):** Stellt den Speicherort für globale Variablen auf einen lokalen oder freigegebenen Ordner ein. Geben Sie den genauen Pfad ein oder klicken Sie auf **Öffnen**, um die Datei zu finden.

Lizenzierte Drucker

> **ANMERKUNG**
> Im Rahmen von Mehrbenutzer-Lizenzen lassen sich Informationen zur Nutzung von lizenzierten Druckern archivieren.

Im Bereich **Lizenzierte Drucker** finden Sie protokollierte Informationen über die Anzahl von Druckern, die in Ihrer Druckumgebung verwendet wurden.

Die Gruppe **Details zu lizenzierten Druckern** zeigt an, wie viele der erlaubten Druckerplätze nach dem Druck auf mehreren Druckern verwendet werden.

- **Anzahl von Druckern, die durch die Lizenz abgedeckt werden:** Anzahl von Druckern, die mit der aktuellen NiceLabel 10Lizenz verwendet werden können.
- **Anzahl von in den letzten 7 Tagen verwendeten Druckern:** Anzahl der Drucker, die innerhalb der letzten 7 Tage mit NiceLabel 10 verwendet wurden.

  > **TIPP**
  > Innerhalb von 7 Tagen ermöglicht die NiceLabel 10 Lizenz nur die Nutzung der angegebenen Anzahl unterschiedlicher Drucker.

> **WARNUNG**
> Die gekaufte Lizenz bestimmt die Anzahl erlaubter Drucker. Wenn Sie diese Anzahl überschreiten, wird eine Warnung angezeigt. Wenn Sie die Anzahl von erlaubten Druckern um das Doppelte überschreiten, können Sie auf neu hinzugefügten Druckern nicht mehr drucken.

Siehe Druckstatus in mehreren Spalten:

- **Drucker:** Name oder Modell des Druckers, der für den Druckauftrag ausgewählt wurde.

  > **ANMERKUNG**
  > Handelt es sich beim verbundenen Drucker um ein gemeinsam genutztes Gerät, wird nur das Modell angezeigt.

- **Speicherort:** Name des Computers, von dem der Druckauftrag gesendet wurde.
- **Anschluss:** Vom Drucker verwendete Schnittstelle.
- **Zuletzt verwendet:** Verstrichene Zeit seit dem letzten Druckauftrag.
- **Reserviert:** Verhindert, dass der Drucker entfernt wird, wenn er mehr als 7 Tage nicht verwendet wurde.

> **ANMERKUNG**
> Wird ein Drucker mehr als 7 Tage lang nicht verwendet, wird er automatisch entfernt, sofern nicht die Option **Reserviert** aktiviert ist.

Der **Berechtigungen**-Bereich ermöglicht es Ihnen, die Druckernutzung am lokalen Rechner zu sperren.

- **Nur reservierte Drucker nutzen:** Wenn diese Option aktiviert ist, dürfen nur reservierte Drucker zum Bearbeiten und Drucken in NiceLabel 10 verwendet werden.

  > **TIPP**
  > Verwenden Sie diese Option, um ein Überschreiten der Anzahl verfügbarer lizenzierter Druckerplätze durch Druck auf unerwünschten Druckern oder mit Print-to-File-Anwendungen zu verhindern. Reservieren Sie Ihre Thermo- oder Laser-Etikettendrucker und beschränken Sie den Druck auf diese Modelle. So stellen Sie kontinuierlichen Etikettendruck mit einer Mehrbenutzer-Lizenz sicher.

  Diese Option kann auch anhand der Datei `product.config`:

  1. Navigieren Sie zum Systemordner.
     %PROGRAMDATA%\NiceLabel\NiceLabel 10
  2. Erstellen Sie eine Sicherungskopie der Datei **`product.config`**.
  3. Öffnen Sie **`product.config`** in einem Text-Editor. Die Datei hat eine XML-Struktur.
  4. Fügen Sie die folgenden Zeilen hinzu:

     ```
     <Configuration>
       <Activation>
         <ReservePrinters>Example Printer Name</ReservePrinters>
       </Activation>
       <Common>
         <General>
           <ShowOnlyReservedPrinters>True</ShowOnlyReservedPrinters>
         </General>
       </Common>
     </Configuration>
     ```

  5. Speichern Sie die Datei. Der `Example Printer` ist reserviert.

Control Center

Der Bereich **Control Center** ermöglicht Ihnen die Aktivierung und Konfiguration der Überwachung von Ereignissen und Druckaufträgen. NiceLabel Control Center ermöglicht zentralisierte Berichterstattung über Ereignisse und Druckaufträge sowie das zentralisierte Speichern globaler Variablen.

ANMERKUNG

Diese Registerkarte ist mit LMS Pro oder LMS Enterprise Lizenzen verfügbar.

Die Gruppe **Adresse** definiert, welcher NiceLabel Control Center Server verwendet werden soll.

- **Control Center Serveradresse:** URL des verbundenen NiceLabel Control Center Servers. Sie können eine Auswahl aus der Liste von automatisch gefundenen Servern im Netzwerk treffen oder manuell eine Serveradresse eingeben.

ANMERKUNG

Lizenzschlüssel auf dem NiceLabel Control Center Server und auf dem Arbeitsplatzrechner müssen übereinstimmen, damit die Verbindung hergestellt werden kann.

Die Gruppe **Ereignismonitor** definiert, welche Arten von Ereignissen vom verbundenen NiceLabel Control Center protokolliert werden.

- **Druck-Ereignisse:** Protokolliert die druck-bezogenen Ereignisse vom Arbeitsplatzrechner.
- **Fehlerereignisse:** Protokolliert alle gemeldeten Fehler.

ANMERKUNG

Standardmäßig werden Druck- und Fehlerereignisse im NiceLabel Control Center protokolliert.

- **Triggeraktivität:** Protokolliert alle ausgelösten Trigger.
- **Trigger-Statusänderungsereignisse:** Protokolliert die Trigger-Statusänderungen, die von den ausgelösten Triggern eingeleitet wurden.

Die Gruppe **Druckauftrag Überwachung** ermöglicht Ihnen die Protokollierung der erledigten und laufenden Druckaufträge in NiceLabel Control Center.

- **Druckauftrag Protokoll auf dem Server aktivieren:** Aktiviert die Protokollierung von Druckaufträgen.
- **Detaillierte Druckerkontrolle:** Ermöglicht die Überwachung der Status, die vom verbundenen Drucker gemeldet werden.

ANMERKUNG

Zwei Voraussetzungen müssen erfüllt sein, damit diese Option verfügbar ist:

- Der Drucker muss bidirektionale Kommunikation unterstützen.
- NiceLabelDer Druckertreiber muss für den Druck verwendet werden.

Automatisierung

Diese Einstellungen legen die erweiterten Funktionen der Anwendung fest.

ANMERKUNG

Die Änderungen werden wirksam, wenn Sie die Anwendung neu starten.

Kommunikation mit dem Dienst

- **Port für die Kommunikation mit dem Dienst:** Automation Manager steuert den Dienst anhand des TCP/IP-Protokolls über den ausgewählten Port. Wenn sich der Standardport nicht für die Nutzung an Ihrem Computer eignet, wählen Sie eine andere Portnummer aus. Achten Sie dabei darauf, dass die gewählte Portnummer nicht bereits durch eine andere Anwendung genutzt wird.

Log

- **Log-Einträge täglich löschen um:** Legt den Beginn des Aufräumprozesses fest. Alte Ereignisse werden aus der Protokolldatenbank entfernt.
- **Log-Einträge löschen, die älter sind als (Tage):** Legt fest, wie lange Ereignisse in der Protokolldatenbank verwahrt werden. Alle Ereignisse, die älter als die angegebene Anzahl von Tagen sind, werden bei jedem Aufräumereignis aus der Datenbank gelöscht.
- **Protokollnachrichten:** Legt den Umfang der Ereignisaufzeichnung im Protokoll fest. Während der Entwicklungs- und Testphasen sollten Sie eventuell eine ausführliche Protokollierung aktivieren. Aktivieren Sie zu diesem Zweck alle Nachrichten, um die Verfolgung der Trigger-Ausführung zu optimieren. Später, in der Produktionsphase, sollten Sie das Ausmaß der Protokollierung einschränken, sodass nur Fehlermeldungen aufgezeichnet werden.

Performance

- 

  PRODUKTEBENEN-INFO

  Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

  Cachedateien aus Dokumentenspeicher und Netzwerkfreigaben: Um die Zeit bis zum ersten gedruckten Etikett und die Gesamtperformance zu steigern, unterstützt NiceLabel Automation das Zwischenspeichern von Dateien. Nachdem Sie Etiketten, Bilder und Datenbankdaten aus Netzwerkfreigaben geladen haben, müssen alle erforderlichen Dateien abgerufen werden, bevor der Druckprozess beginnen kann.

  TIPP

  Wenn Sie lokales Zwischenspeichern aktivieren, werden die Auswirkungen der Netzwerklatenz gemindert, da Etiketten- und Bilddateien von Ihrem lokalen Datenträger geladen werden.

AutomationDienst nutzt den folgenden lokalen Ordner, um die entfernten Dateien zwischenzuspeichern: `%PROGRAMDATA%\NiceLabel\NiceLabel 10\FileCache`.

- **Aktualisierungsintervall (Minuten):** Legt das Zeitintervall fest, in dem die Dateien im Zwischenspeicher mit den Dateien im Ursprungsordner synchronisiert werden. Dies ist das Zeitlimit, innerhalb dessen das Systemen eventuell Versionen verwenden kann, die nicht aktuell sind.
- **Ungenutzte Dateien im Zwischenspeicher löschen nach (Tagen):** legt das Zeitintervall fest, nach dem alle Dateien aus dem Zwischenspeicher entfernt werden.

> **ANMERKUNG**
> Beim Zwischenspeichern werden Etiketten- und Bilddateiformate unterstützt. Starten Sie nach Aktivierung des Zwischenspeicherns den Automation Dienst neu, um die Änderungen wirksam zu machen.

- **Ordner aus dem Dokumentenspeicher vorab in den Zwischenspeicher laden** ermöglicht es Ihnen, die Dateien aus Control Center Dokumentenspeicher-Ordnern auf Ihrem Computer lokal zwischenzuspeichern. Wenn das Vorab-Laden in den Zwischenspeicher aktiviert ist, wird der Inhalt des lokalen Zwischenspeichers weiterhin mit den ausgewählten Dokumentenspeicher-Ordnern synchronisiert.

> **ANMERKUNG**
> Im Vergleich zum Zwischenspeichern mindert das Vorab-Zwischenspeichern die Druckzeit für Ihr erstes gedrucktes Etikett.

Fügen Sie jeden Dokumentenspeicher-Ordner in einer separaten Zeile hinzu.

```
/Labels/Folder1

/Labels/Folder2
```

> **ANMERKUNG**
> Ihr Automation Builder muss mit dem Control Center verbunden sein, um die Offline-Synchronisierung von Dateien im Zwischenspeicher zu ermöglichen.

Cluster-Unterstützung

> **PRODUKTEBENEN-INFO**
> Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

Diese Einstellung aktiviert Unterstützung für Hochverfügbarkeits-Cluster (Failover) in NiceLabel Automation. Wählen Sie den Ordner aus, den beide Knoten im Cluster verwenden sollen, um Daten über Echtzeit-Triggerstatus auszutauschen.

Designer

Im **Designer**-Bereich können Sie das Öffnungsverhalten von NiceLabel 10 konfigurieren.

- **Jedes Dokument in einem eigenen Fenster anzeigen:** Sofern aktiviert, werden weitere geöffnete Dokumente in separaten Fenstern von NiceLabel 10 angezeigt. Dies gilt sowohl für neu erstellte als auch für vorhandene Dokumente.
  Wenn Sie entscheiden, diese Option zu deaktivieren, werden zusätzlich geöffnete Dokumente innerhalb der aktuell aktiven Instanz von NiceLabel 10 angezeigt.
- **Quelle für Druckereinstellungen** ermöglicht Ihnen, die Quelle für die Druckereinstellungen auszuwählen.
  - **Druckereinstellungen aus dem Druckertreiber nutzen:** Wählen Sie diese Option aus, wenn Sie das Drucken anhand der Druckertreiber-Einstellungen bevorzugen. Mit dieser Option können Sie die Druckereinstellungen in Ihrer Arbeitsumgebung standardisieren.
  - **Im Etikett gespeicherte benutzerdefinierte Druckereinstellungen verwenden:** Benutzer können für jedes Etikett eigene Druckereinstellungen festlegen und speichern. Wählen Sie diese Option, um beim Drucken eigene Einstellungen für Ihre Etiketten zu verwenden.

1.6.6. Über

Der Dialog „Über“ bietet Informationen zu Ihrer NiceLabel Produktlizenz, ermöglicht den Lizenzkauf (im Testmodus) und die Aktivierung, bietet Details zur Software und versetzt Sie in die Lage, die Produktebene von NiceLabel 10 zu wechseln.

Die Gruppe **Lizenzinformation** beinhaltet:

- **Testmodus-Dauer:** Anzahl der verbleibenden Tage für die Evaluation des Produkts. Nach dem Kauf und der Aktivierung einer Produktlizenz ist dieser Abschnitt nicht mehr sichtbar.
- **Lizenz kaufen:** Diese Schaltfläche leitet Sie zum NiceLabel Online-Store.
- **Lizenz aktivieren:** Diese Schaltfläche öffnet den Lizenzaktivierungs-Dialog von NiceLabel 10. Im NiceLabel 10 Installationshandbuch finden Sie nähere Angaben zum Lizenzaktivierungsprozess. Nach Aktivieren der Lizenz heißt diese Schaltfläche **Lizenz deaktivieren** – nachdem Sie sie angeklickt und danach die Deaktivierung bestätigt haben, ist Ihre Kopie von NiceLabel 10 nicht mehr aktiviert.
- **Produktebene ändern:** Öffnet den Dialog zur Auswahl der Produktebene. Im Testmodus können Sie alle Produktebenen auswählen und testen. Nach Aktivierung Ihrer Lizenz können Sie die Produktebene nur auf niedrigere Ebenen ändern.

ANMERKUNG

Änderungen der Produktebene werden nach Neustart der Anwendung wirksam.

- **Upgrade-Lizenz:** Öffnet den Dialog für Upgrades der Produktebene. Im NiceLabel 10 Installationshandbuch finden Sie nähere Angaben zum Lizenz-Upgradeprozess.

Die Gruppe **Softwareinformation** enthält Informationen zur installierten Softwareversion und der Build-Nummer.

1.6.6.1. Verlust Ihrer Label Cloud-Verbindung

Wenn Ihr Automation Manager bei der Label Cloud angemeldet ist und die Internetverbindung unterbrochen wird, müssen Sie die Verbindung innerhalb von fünf Tagen wiederherstellen. Ohne erneute Verbindung mit Ihrer Label Cloud wird Automation Manager automatisch geschlossen.

Nach Verlust Ihrer Internetverbindung wird nach 5 Tagen eine Warnung angezeigt, wenn Ihr Computer offline bleibt.Automation Manager wird 5 Minuten nach Anzeige der Warnung beendet.

Öffnen Sie nach Wiederherstellung der Internetverbindung Automation Builder oder Automation Manager und melden Sie sich bei Label Cloud an. Dadurch wird Ihre Kopie wieder aktiviert.

WARNUNG

Speichern Sie Ihre Arbeit an einem Offline-Speicherort (Ihrem Computer), damit keine Änderungen verloren gehen.

2. Informationen zu Filtern

NiceLabel Automation nutzt Filter, um die Struktur der Daten zu definieren, die von Triggern empfangen werden. Immer wenn ein Trigger Daten empfängt, werden sie von einem oder mehreren Filtern geparst. Dabei werden relevante Werte für Ihre Konfiguration extrahiert. Jeder Filter enthält Regeln zur Erkennung von Feldern in den empfangenen Daten.

ANMERKUNG

Als Ergebnis stellt der Filter eine Liste von Feldern und deren Werten bereit (**`Name:Wert`**-Paare).

Filtertypen

Weitere Informationen finden Sie in den Abschnitten Filter für strukturierten Text, Filter für unstrukturierte Daten, XML-Filter, und JSON-Filter.

Datenstruktur

Die Filterkomplexität hängt von der Datenstruktur ab. Bei Dateien mit strukturierten Daten, wie z. B. CSV- oder XML-Dateien, ist die Datenextraktion einfach. In solchen Fällen werden die Feldnamen bereits durch die Daten definiert. Die Extraktion von **`Name:Wert`**-Paaren geht schnell. Im Fall von unstrukturierten Daten braucht man mehr Zeit für die Definition der Extraktionsregeln. Sie könnten auf solche Daten stoßen, wenn Sie Dokumente und Berichte aus Altsystemen, abgefangene Kommunikation zwischen Geräten oder erfasste Druckströme exportieren.

Der Filter definiert eine Liste von Feldern, die aus den eingehenden Daten extrahiert werden, sobald Sie den Filter ausführen.

NiceLabel Automation unterstützt verschiedene Typen von Eingabedaten, die allesamt von einem der unterstützten Filtertypen geparst werden können. Wählen Sie den richtigen Filter für den jeweiligen Eingabedatentyp aus. Beispielsweise würden Sie für eingehende CSV-Daten den **Filter für strukturierten Text**, für eingehende JSON-Daten den **JSON-Filter** und für XML-Daten den **XML-Filter** verwenden. Für alle Arten von unstrukturierten Daten würden Sie den **Filter für unstrukturierte Daten** verwenden. Weitere Informationen finden Sie im Abschnitt Informationen zu Datenstrukturen.

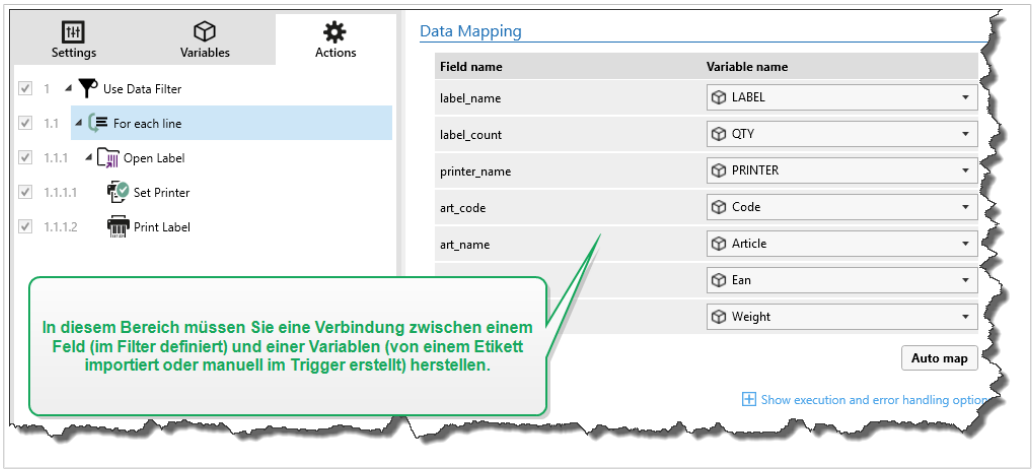
Daten extrahieren

Ein Filter ist ein Satz von Regeln und führt die Extraktion nicht selbst aus. Um den Filter auszuführen, verwenden Sie die Aktion Datenfilter verwenden. Diese Aktion wendet Filterregeln auf die jeweiligen Daten an und extrahiert die Werte.

Jeder Trigger-Typ kann beliebig viele „Datenfilter verwenden“-Aktionen ausführen. Wenn Sie zusammengesetzte Daten erhalten, die nicht von einem einzigen Filter geparst werden können, können Sie mehrere Filter definieren und ihre Regeln anhand einer Abfolge von „Datenfilter verwenden“-Aktionen ausführen. Danach können Sie die extrahierten Werte aus allen Aktionen auf demselben Etikett verwenden.

Felder Variablen zuordnen

Um die extrahierten Werte zu nutzen, müssen Sie sie in Variablen speichern. Die Aktion „Datenfilter verwenden“ tut beides – sie extrahiert Werte und speichert sie in Variablen. Um diesen Prozess zu konfigurieren, ordnen Sie jede Variable dem entsprechenden Feld zu. Der Wert des Feldes wird dann in der zugeordneten Variablen gespeichert.



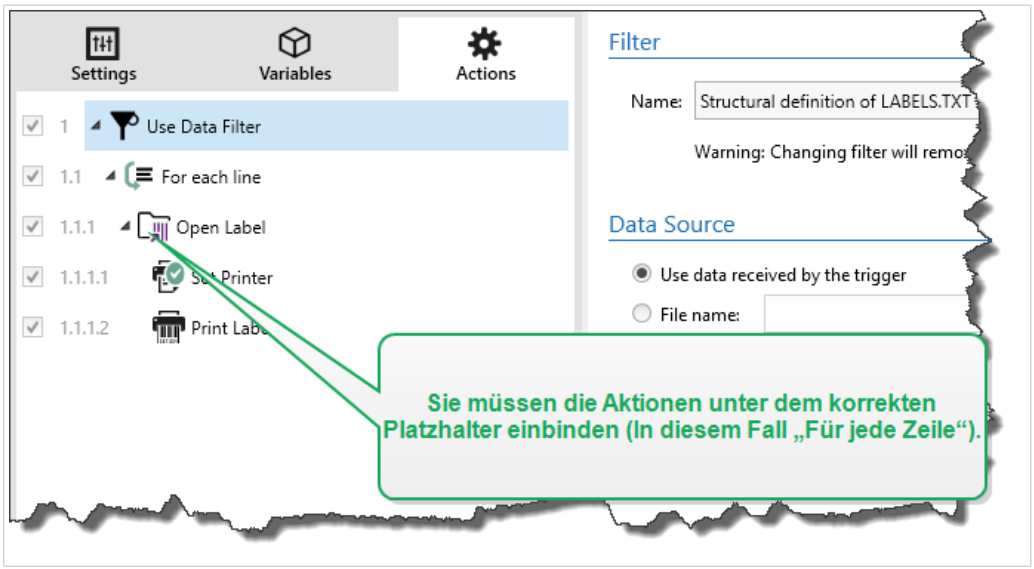
TIPP

Es empfiehlt sich, Felder und Variablen mit identischen Namen zu definieren. In diesem Fall verbindet die automatische Zuordnung Variablen mit den gleichnamigen Feldern, wodurch die manuelle Zuordnung überflüssig wird.

Automatische Zuordnung ist für alle unterstützten Filtertypen verfügbar. Wenn die automatische Zuordnung aktiviert ist, extrahiert die „Datenfilter verwenden“-Aktion Werte und ordnet sie automatisch den Variablen zu, deren Namen mit denen der jeweiligen Felder identisch sind. Weitere Informationen finden Sie in den Abschnitten Dynamische Struktur aktivieren für den Filter für strukturierten Text, Zuweisungsbereiche definieren für den Filter für unstrukturierte Daten und „Zuweisungsbereiche definieren“ für XML- oder JSON-Filter.

Aktionen mit extrahierten Daten ausführen

Für gewöhnlich möchten Sie bestimmte Aktionen anhand der extrahierten Daten ausführen, z. B. **Etikett öffnen**, **Etikett drucken** oder eine der Aktionen für ausgehende Verbindungen. Es ist sehr wichtig, solche Aktionen unter der **Datenfilter verwenden**-Aktion einzubinden. So stellen Sie sicher, dass die eingebundenen Aktionen für jede Datenextraktion ausgeführt werden.



Beispiel

Wenn Sie eine CSV-Datei mit fünf Zeilen haben, werden auch die eingebundenen Aktionen fünfmal ausgeführt, einmal für jede Datenextraktion. Sind die Aktionen nicht eingebunden, werden sie nur einmal ausgeführt und enthalten nur Daten aus der letzten Datenextraktion. Beim obigen Beispiel würde die 5. CSV-Zeile gedruckt, die ersten vier Zeilen nicht. Falls Sie Unterbereiche nutzen, müssen Sie sicherstellen, dass Sie Ihre Aktion unter dem richtigen Platzhalter einbinden.

2.1. Filter für strukturierten Text konfigurieren

2.1.1. Filter für strukturierten Text

Um mehr über Filter im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Filtern.

Verwenden Sie diesen Filter, wenn Sie eine strukturierte Textdatei erhalten, in der die Felder anhand einer der folgenden Methoden identifiziert werden können:

- **Felder werden durch ein Zeichen getrennt:** Übliche Trennzeichen sind Komma oder Semikolon. CSV (durch Kommas getrennte Werte) ist ein typisches Beispiel für eine getrennte Datei.
- **Felder enthalten eine feste Anzahl von Zeichen:** Anders gesagt, Felder werden durch Spalten mit fester Breite definiert.

Beispiele für strukturierte Textdaten finden Sie in Abschnitt Textdatenbank.

Struktur definieren

Um die Struktur einer Textdatei zu definieren, haben Sie folgende Optionen:

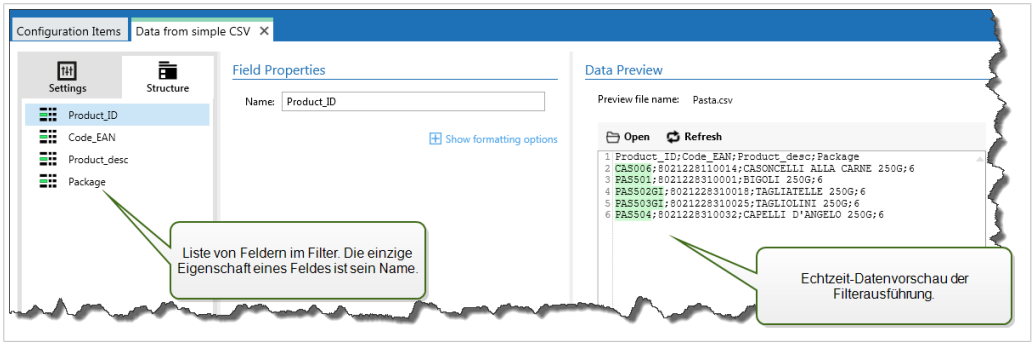
- **Struktur anhand des Textdatei-Assistenten importieren:** Klicken Sie in diesem Fall auf die Schaltfläche **Datenstruktur importieren** in der Menüleiste und folgen Sie den Anweisungen auf dem Bildschirm. Nach Beenden des Assistenten sind die Art der Textdatenbank sowie alle Felder definiert. Falls die erste Datenzeile die Feldnamen enthält, kann der Assistent sie importieren. Dies ist die empfohlene Methode, wenn der Trigger immer Daten empfängt, deren Struktur sich nicht ändert.
- **Manuelle Definition der Felder:** In diesem Fall müssen Sie den Typ der Daten (getrennte Felder oder Felder mit fester Länge) und dann die Feldnamen manuell definieren. Weitere Informationen finden Sie im Abschnitt Felder Definieren .
- **Dynamisches Auslesen der Felder:** In diesem Fall empfängt der Trigger möglicherweise Daten, die auf andere Art strukturiert sind. An Beispiel dafür sind neue Feldnamen – eine dynamische Struktur macht das Aktualisieren des Filters für jede Strukturelle Änderung überflüssig. Die Unterstützung für dynamische Struktur liest automatisch alle Datenfelder aus, unabhängig davon, ab ein neues Feld vorhanden ist oder einige der alten Felder fehlen. Danach ordnet sie sie automatisch den Variablen mit identischen Namen zu. Weitere Informationen finden Sie im Abschnitt Dynamische Struktur aktivieren.

Der „Datenvorschau“-Bereich vereinfacht die Konfiguration. Das Ergebnis einer definierten Filterregel wird bei jeder Konfigurationsänderung im Vorschaubereich hervorgehoben. Mithilfe der Datenvorschau können Sie prüfen, welche Daten mit jeder einzelnen Rolle extrahiert werden.

2.1.2. Felder definieren

Bei strukturierten Textdateien ist die Definition von Feldern sehr unkompliziert. Es gibt zwei Optionen:

- **Felder werden durch Trennzeichen definiert:** In diesem Fall haben Sie ein Trennzeichen wie Komma oder Semikolon zwischen den Feldern. Sie müssen lediglich die Feldnamen in der Reihenfolge definieren, in der sie in den vom Trigger empfangenen Daten auftreten.
- **Fester Spaltenbreite:** In diesem Fall müssen Sie lediglich Folgendes definieren: die Feldnamen in der Reihenfolge, in der sie in den vom Trigger empfangenen Daten auftreten, sowie die Anzahl von Zeichen innerhalb eines Feldes. Damit geben Sie vor, wie viele Zeichen als Daten des jeweiligen Feldes ausgelesen werden.



Datenvorschau

Dieser Bereich zeigt eine Vorschau der Felddefinition an. Wenn das definierte Element ausgewählt wird, zeigt die Vorschau dessen Position in den Vorschaudaten an.

- **Name der Vorschaudatei:** Gibt die Datei an, welche die Beispieldaten enthält, die durch den Filter geparst werden sollen. Die Vorschaudatei wird aus der Filterdefinition kopiert. Wenn Sie den Namen der Vorschaudatei ändern, wird der neue Name gespeichert.
- **Öffnen:** Wählt eine andere Datei aus, auf die die Filterregeln angewandt werden sollen.
- **Aktualisieren:** Wendet die Filterregeln erneut auf den Inhalt der Vorschaudatei an.Automation Aktualisiert den „Datenvorschau“-Bereich mit dem Ergebnis.

Formatierungsoptionen

Dieser Abschnitt definiert Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden in der Reihenfolge ausgeführt, die in der Benutzeroberfläche ausgewählt ist – von oben nach unten.

- **Leerzeichen am Anfang löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Anfang einer Zeichenfolge.
- **Leerzeichen am Ende löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Ende einer Zeichenfolge.
- **Eröffnungs- und Abschlusszeichen löschen:** Löscht die erste Instanz der ausgewählten Eröffnungs- und Abschlusszeichen, die in einer Zeichenfolge enthalten sind.

Beispiel

Wenn Sie „{“ als Eröffnungszeichen und „}“ als Abschlusszeichen verwenden, wird die Eingabe-Zeichenfolge **`{{Auswahl}}`** in **`{Auswahl}`** konvertiert.

- **Suchen und ersetzen:** Führt anhand der angegebenen Werte für *Finde* und *Ersetzen durch* eine Standardfunktion für Suchen und Ersetzen durch. Sie können auch reguläre Ausdrücke verwenden.

  ANMERKUNG

  Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. Nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im Knowledge Base-Artikel.

- **Nicht-druckbare Zeichen durch Leerzeichen ersetzen:** Ersetzt alle Steuerzeichen in einer Zeichenfolge durch Leerzeichen (ASCII-Dezimalwert 32). Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Nicht druckbare Zeichen löschen:** Löscht alle Steuerzeichen in der Zeichenfolge. Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Sonderzeichen dekodieren:** Sonderzeichen (oder Steuercodes) sind Zeichen, die keine Entsprechung auf der Tastatur haben, z. B. Wagenrücklauf oder Zeilenvorschub. verwendet eine menschenlesbare Codierung für solche Zeichen, beispielsweise <CR> für Wagenrücklauf und <LF> für Zeilenvorschub. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen (Steuercodes).
  Diese Option konvertiert Sonderzeichen aus der Syntax in tatsächliche Binärzeichen.

Beispiel

Wenn Sie die Datenfolge „<CR><LF>“ empfangen, fasst sie als reine Zeichenfolge aus 8 Zeichen auf. Um die empfangenen Daten als zwei Binärzeichen zu erkennen, **CR** (Wagenrücklauf – ASCII-Code 13) und **LF** (Zeilenvorschub – ASCII-Code 10), müssen Sie diese Option aktivieren.

- **Suchen und Löschen von allem vor:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab Beginn der Daten bis zu der Zeichenfolge. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab der Zeichenfolge bis zum Ende der Daten. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Großschreibung ändern**: Ändert alle Zeichen in Ihren Zeichenfolgen zu Groß- oder Kleinschreibung.

2.1.3. Dynamische Struktur aktivieren

Der Filter für strukturierten Text kann Felder und ihre Werte in den empfangenen Daten automatisch erkennen. Daher ist eine manuelle *Variable-zu-Feld*-Zuordnung nicht notwendig.

Die Funktion **Dynamische Struktur** ist nützlich, wenn der Trigger Daten mit sich ändernder Struktur empfängt. In solchen Fällen bleibt die Haupt-Datenstruktur unverändert (z. B. Felder werden durch Kommas getrennt) oder dieselbe Struktur wird beibehalten, aber **die Reihenfolge** und/oder **die Anzahl** von Feldern ändert sich. Es könnten neue Felder hinzukommen oder alte Felder könnten nicht mehr verfügbar sein. Aufgrund der aktivierten **Dynamischen Struktur** erkennt der Filter die Struktur der empfangenen Datei automatisch. Gleichzeitig liest der Filter Feldnamen und Werte (`Name-Wert`-Paare) aus den Daten. Dadurch müssen Felder nicht manuell Variablen zugeordnet werden.

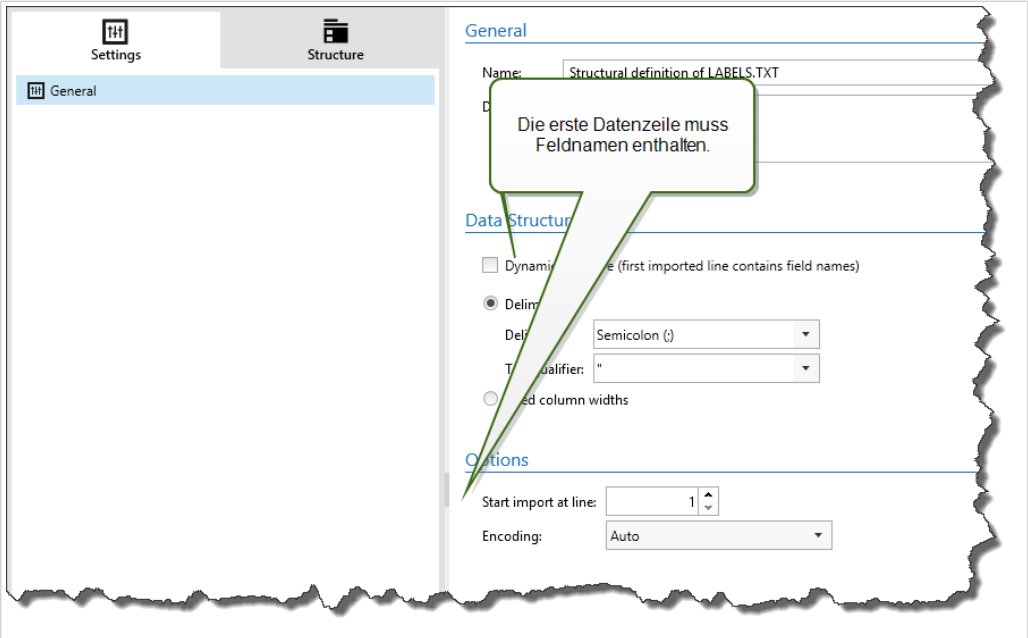
Die Aktion Datenfilter verwenden bietet keine Zuordnungsmöglichkeiten, da sie die Zuordnung dynamisch vornimmt. Sie müssen nicht einmal die Etikettenvariablen in der Trigger-Konfiguration festlegen. Die Aktion ordnet Feldwerte den gleichnamigen Etikettenvariablen zu, ohne dass die Variablen aus dem Etikett importiert werden müssen. Diese Regel gilt jedoch nur für die Aktion Etikett drucken. Wenn Sie die Feldwerte in einer anderen Aktion verwenden möchten, müssen Sie Variablen im Trigger definieren, dabei aber die automatische Zuordnung von *Variablen zu Feldern* beibehalten.

ANMERKUNG

Wenn ein Feld in den Eingabedaten keine entsprechende Variable auf dem Etikett hat, tritt kein Fehler auf. Ignoriert die fehlenden Variablen, ohne eine Meldung auszugeben.

Dynamische Struktur konfigurieren

Um die dynamische Struktur zu konfigurieren, aktivieren Sie die Option **Dynamische Struktur** in den Eigenschaften des Filters für strukturierten Text.



- Die erste Datenzeile muss Feldnamen enthalten.
- Die Zeile, die Sie für **Import bei folgender Zeile beginnen** auswählen, muss die Feldnamen enthalten (für gewöhnlich die erste Zeile in den Daten).
- Die Datenstruktur muss eine Trennung (durch Trennzeichen oder feste Spaltenbreite) aufweisen.
- Falls nötig, können Sie die Daten formatieren.

Formatierungsoptionen

Dieser Abschnitt definiert Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden in der Reihenfolge ausgeführt, die in der Benutzeroberfläche ausgewählt ist – von oben nach unten.

- **Leerzeichen am Anfang löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Anfang einer Zeichenfolge.
- **Leerzeichen am Ende löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Ende einer Zeichenfolge.
- **Eröffnungs- und Abschlusszeichen löschen:** Löscht die erste Instanz der ausgewählten Eröffnungs- und Abschlusszeichen, die in einer Zeichenfolge enthalten sind.

Beispiel

Wenn Sie „{“ als Eröffnungszeichen und „}“ als Abschlusszeichen verwenden, wird die Eingabe-Zeichenfolge `{{Auswahl}}` in `{Auswahl}` konvertiert.

- **Suchen und ersetzen:** Führt anhand der angegebenen Werte für *Finde* und *Ersetzen durch* eine Standardfunktion für Suchen und Ersetzen durch. Sie können auch reguläre Ausdrücke verwenden.

  > **ANMERKUNG**
  > Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. Nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im Knowledge Base-Artikel.

- **Nicht-druckbare Zeichen durch Leerzeichen ersetzen:** Ersetzt alle Steuerzeichen in einer Zeichenfolge durch Leerzeichen (ASCII-Dezimalwert 32). Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Nicht druckbare Zeichen löschen:** Löscht alle Steuerzeichen in der Zeichenfolge. Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Sonderzeichen dekodieren:** Sonderzeichen (oder Steuercodes) sind Zeichen, die keine Entsprechung auf der Tastatur haben, z. B. Wagenrücklauf oder Zeilenvorschub. verwendet eine menschenlesbare Codierung für solche Zeichen, beispielsweise <CR> für Wagenrücklauf und <LF> für Zeilenvorschub. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen (Steuercodes).
  Diese Option konvertiert Sonderzeichen aus der Syntax in tatsächliche Binärzeichen.

Beispiel

Wenn Sie die Datenfolge „<CR><LF>“ empfangen, fasst sie als reine Zeichenfolge aus 8 Zeichen auf. Um die empfangenen Daten als zwei Binärzeichen zu erkennen, **CR** (Wagenrücklauf – ASCII-Code 13) und **LF** (Zeilenvorschub – ASCII-Code 10), müssen Sie diese Option aktivieren.

- **Suchen und Löschen von allem vor:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab Beginn der Daten bis zu der Zeichenfolge. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab der Zeichenfolge bis zum Ende der Daten. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Großschreibung ändern**: Ändert alle Zeichen in Ihren Zeichenfolgen zu Groß- oder Kleinschreibung.

2.2. Filter für unstrukturierte Daten konfigurieren

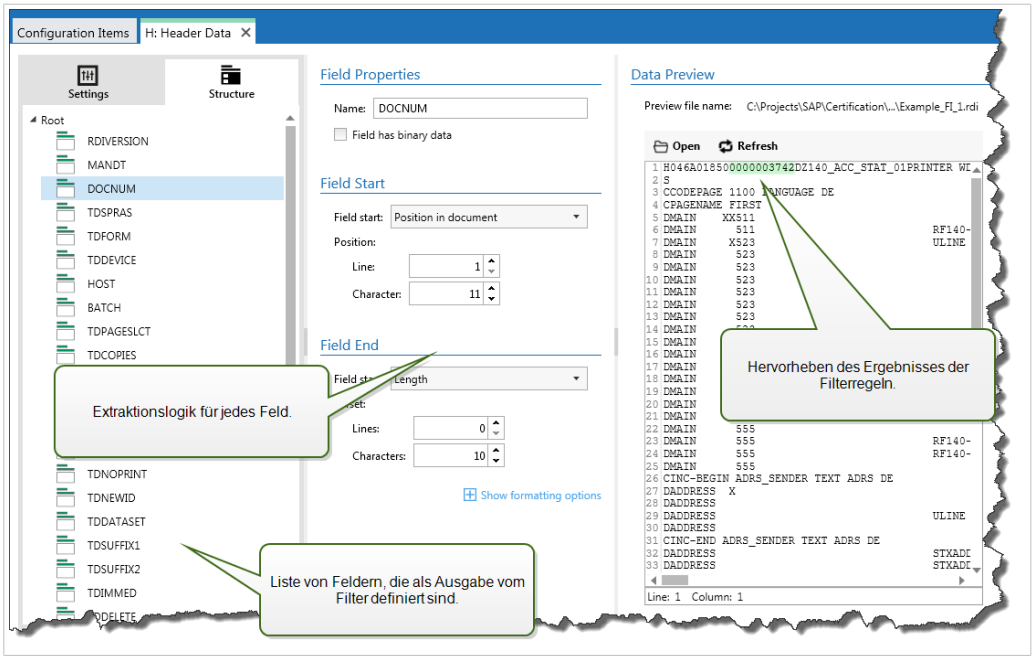
2.2.1. Filter für unstrukturierte Daten

Um mehr über Filter im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Filtern.

Verwenden Sie diesen Filter, wenn ein Trigger unstrukturierte Daten empfängt, z. B. Dokumente und Berichte aus einem alten System, abgefangene Kommunikation zwischen Geräten und erfasste

Druckströme. Der Filter ermöglicht es Ihnen, einzelne Felder, Felder in sich wiederholenden Unterbereichen und sogar **`Name-Wert`**-Paare zu extrahieren.

Beispiele für strukturierte Textdaten finden Sie in den Abschnitten Altdaten, Zusammengesetzte CSV-Dateien und Binärdateien.



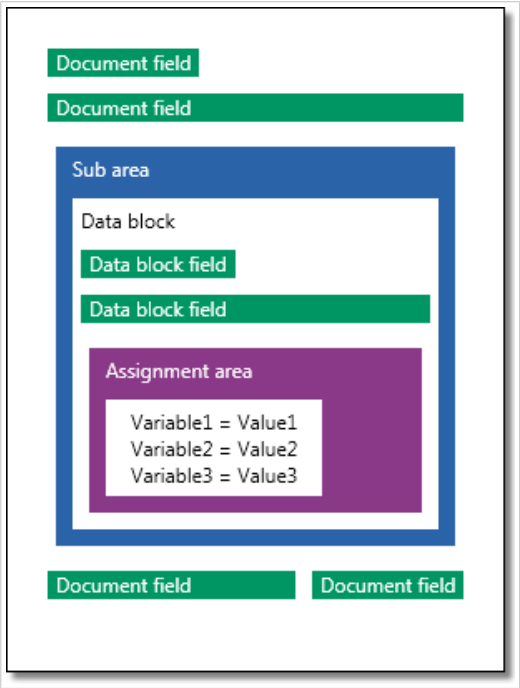
Struktur definieren

Folgende Elemente können Sie zur Konfiguration des Filters verwenden:

- **Feld:** Gibt die Position von Felddaten zwischen der Start- und Endposition von Feldern an. Es gibt verschiedene Möglichkeiten zur Angabe der Feldposition, die von einer festen Codierung der Position bis hin zur Aktivierung relativer Positionierungen reichen. In der Aktion Datenfilter verwenden müssen Sie die definierten Felder den jeweiligen Variablen zuordnen. Weitere Informationen finden Sie im Abschnitt Felder Definieren .
- **Unterbereich:** Gibt die Position sich wiederholender Daten an. Jeder Unterbereich definiert mindestens einen Datenblock, der wiederum Daten für Etiketten enthält. Es können Unterbereiche in Unterbereichen definiert werden, was die Vorgabe komplexer Strukturen ermöglicht. Sie können Felder innerhalb jedes Datenblocks definieren. In der Aktion müssen Sie die definierten Felder den jeweiligen Variablen zuordnen. Für jeden Unterbereich definiert Automation eine neue Platzhalterebene innerhalb der Aktion „Datenfilter verwenden“, sodass Sie Feldern dieser Ebene Variablen zuordnen können. Weitere Informationen finden Sie im Abschnitt Unterbereiche definieren.
- **Zuweisungsbereich:** Gibt die Position sich wiederholender Daten an, welche die **`Name-Wert`**-Paare enthalten. Automation liest Feldnamen und ihre Werte gleichzeitig aus. Automation Auch die Zuordnung zu Variablen erfolgt in automatisch. Verwenden Sie diese Funktion, um den Filter für die sich ändernden Eingangsdaten einzurichten; so können Sie Wartungsaufwand vermeiden. Sie können den Zuweisungsbereich auf der Stammebene des Dokuments oder innerhalb des Unterbereichs festlegen. Weitere Informationen finden Sie im Abschnitt Zuweisungsbereiche definieren.

Der „Datenvorschau“-Bereich vereinfacht die Konfiguration. Das Ergebnis einer definierten Filterregeln wird bei jeder Konfigurationsänderung im Vorschaubereich hervorgehoben. So können Sie sehen, welche Daten durch die einzelnen Regeln extrahiert würden.

Die Felder können auf Stammebene als Dokumentfelder definiert werden. Die Felder können innerhalb eines Datenblocks definiert werden. Die **`Name-Wert`**-Paare können innerhalb des Zuweisungsbereichs definiert werden.



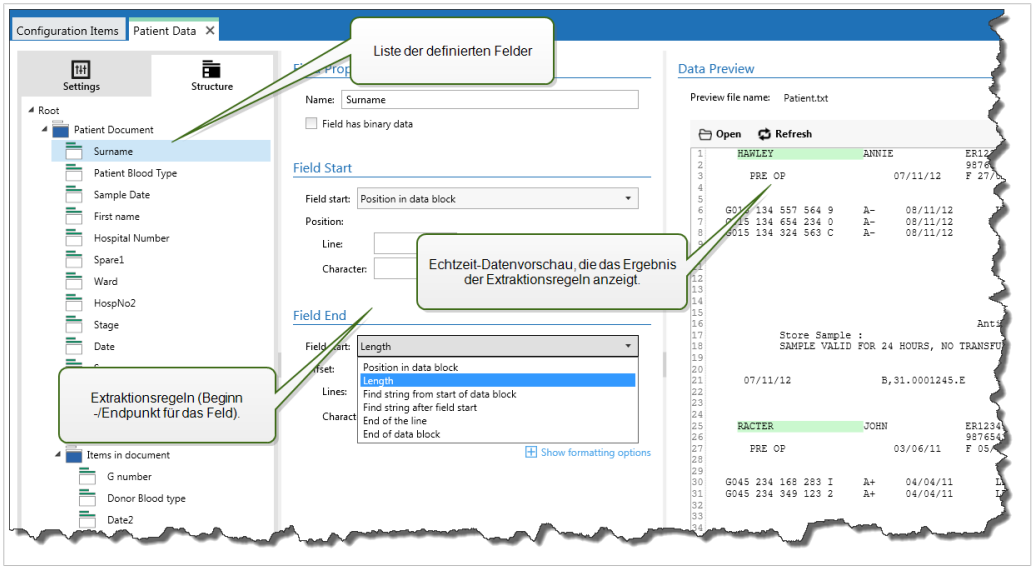
Allgemein

In diesem Abschnitt werden allgemeine Eigenschaften des Filters für unstrukturierte Daten definiert.

- **Name:** Gibt den Filternamen an. Verwenden Sie einen anschaulichen Namen, der die Rolle des Filters in einer Konfiguration beschreibt. Sie können ihn jederzeit ändern.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Filter zu beschreiben. Sie können hier eine kurze Erklärung zur Funktionsweise des Filters eingeben.
- **Codierung:** Gibt die Datencodierung an, mit der dieser Filter arbeitet.
- **Leere Zeilen in Datenblöcken ignorieren:** Legt fest, dass kein Fehler gemeldet wird, wenn der Filter leere Feldwerte aus den Datenblöcken extrahiert.

2.2.2. Felder definieren

Nachdem Sie ein Feld definiert haben, müssen Sie seinen Namen sowie Regeln für die Extraktion von Feldwerten aus den Daten angeben. Bei Ausführung des Filters werden die Extraktionsregeln auf die Eingangsdaten angewendet und ordnen das Ergebnis dem Feld zu.



Feldeigenschaften

- **Name:** Gibt den eindeutigen Namen des Feldes an.
- **Feld enthält binäre Daten:** Gibt an, dass das Feld binäre Daten enthält. Aktivieren Sie diese Option nur, wenn Sie wirklich davon ausgehen, binäre Daten zu erhalten.

Feldanfang

- **Position im Dokument:** Fest codierte Position in den Daten, die den Anfangs-/Endpunkt bestimmt. Der Koordinatenursprung ist in der oberen linken Ecke. Das Zeichen an der definierten Position wird in die extrahierten Daten eingeschlossen.
- **Ende des Dokuments:** Der Start-/Endpunkt befindet sich am Ende des Dokuments. Sie können auch einen Versatz ab Ende des Dokuments um eine bestimmte Anzahl von Zeilen und/oder Zeichen definieren.
- **Zeichenfolge ab dem Beginn des Dokuments finden:** Die Position der gesuchten Zeichenfolge bestimmt den Anfangs-/Endpunkt. Nachdem Automation die gewünschte Zeichenfolge gefunden hat, bestimmt das erste auf sie folgende Zeichen den Start-/Endpunkt. Die gesuchte Zeichenfolge wird nicht in die extrahierten Daten eingeschlossen. Die Standardsuche berücksichtigt Groß-/Kleinschreibung.
    - **Suche bei absoluter Position beginnen:** Sie können eine Feinanpassung der Suche vornehmen, indem Sie die Startposition vom Datenbeginn (Position 1,1) aus versetzen. Verwenden Sie diese Funktion, um die Suche nicht am Datenbeginn zu starten.
    - **Vorkommen:** Gibt an, welche Instanz der gesuchten Zeichenfolge verwendet werden soll. Verwenden Sie diese Option, wenn Sie die Start-/Endposition nicht nach der ersten gefundenen Instanz der Zeichenfolge festlegen möchten.
    - **Versatz ab Zeichenfolge:** Gibt den positiven oder negativen Versatz nach der gesuchten Zeichenfolge an.

Beispiel

Sie legen der Versatz so fest, dass die gesuchte Zeichenfolge in den extrahierten Daten enthalten ist.

Feldende

- **Position im Dokument:** Fest codierte Position in den Daten, die den Anfangs-/Endpunkt bestimmt. Der Koordinatenursprung ist in der oberen linken Ecke. Das Zeichen an der definierten Position wird in die extrahierten Daten eingeschlossen.
- **Ende des Dokuments:** Der Start-/Endpunkt befindet sich am Ende des Dokuments. Sie können auch einen Versatz ab Ende des Dokuments um eine bestimmte Anzahl von Zeilen und/oder Zeichen definieren.
- **Zeichenfolge ab dem Beginn des Dokuments finden:** Die Position der gesuchten Zeichenfolge bestimmt den Anfangs-/Endpunkt. Nachdem Automation die gewünschte Zeichenfolge gefunden hat, bestimmt das erste auf sie folgende Zeichen den Start-/Endpunkt. Die gesuchte Zeichenfolge wird nicht in die extrahierten Daten eingeschlossen. Die Standardsuche berücksichtigt Groß-/Kleinschreibung.
    - **Suche bei absoluter Position beginnen:** Sie können eine Feinanpassung der Suche vornehmen, indem Sie die Startposition vom Datenbeginn (Position 1,1) aus versetzen. Verwenden Sie diese Funktion, um die Suche nicht am Datenbeginn zu starten.
    - **Vorkommen:** Gibt an, welche Instanz der gesuchten Zeichenfolge verwendet werden soll. Verwenden Sie diese Option, wenn Sie die Start-/Endposition nicht nach der ersten gefundenen Instanz der Zeichenfolge festlegen möchten.
    - **Versatz ab Zeichenfolge:** Gibt den positiven oder negativen Versatz nach der gesuchten Zeichenfolge an.

    ### Beispiel

    Sie legen der Versatz so fest, dass die gesuchte Zeichenfolge in den extrahierten Daten enthalten ist.

- **Zeichenfolge nach Feldbeginn finden:** Der Anfangs-/Endpunkt wird durch die Position der gesuchten Zeichenfolge gemäß der Option **Zeichenfolge ab dem Beginn des Dokuments finden** festgelegt, aber die Suche beginnt nach der Anfangsposition des Felds/Bereichs, nicht am Anfang der Daten.
- **Länge:** Gibt die Länge der Daten in Zeilen und/oder Zeichen an. Die angegebene Anzahl von Zeilen und/oder Zeichen wird ab der Startposition extrahiert.
- **Ende der Zeile:** Legt fest, dass die Daten ab der Startposition bis zum Ende derselben Zeile extrahiert werden sollen. Sie können einen negativen Versatz ab dem Zeilenende angeben.

Formatierungsoptionen

Dieser Abschnitt definiert Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden in der Reihenfolge ausgeführt, die in der Benutzeroberfläche ausgewählt ist – von oben nach unten.

- **Leerzeichen am Anfang löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Anfang einer Zeichenfolge.
- **Leerzeichen am Ende löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Ende einer Zeichenfolge.
- **Eröffnungs- und Abschlusszeichen löschen:** Löscht die erste Instanz der ausgewählten Eröffnungs- und Abschlusszeichen, die in einer Zeichenfolge enthalten sind.

Beispiel

Wenn Sie „{“ als Eröffnungszeichen und „}“ als Abschlusszeichen verwenden, wird die Eingabe-Zeichenfolge **`{{Auswahl}}`** in **`{Auswahl}`** konvertiert.

- **Suchen und ersetzen:** Führt anhand der angegebenen Werte für *Finde* und *Ersetzen durch* eine Standardfunktion für Suchen und Ersetzen durch. Sie können auch reguläre Ausdrücke verwenden.

  ANMERKUNG

  Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. Nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im Knowledge Base-Artikel.

- **Nicht-druckbare Zeichen durch Leerzeichen ersetzen:** Ersetzt alle Steuerzeichen in einer Zeichenfolge durch Leerzeichen (ASCII-Dezimalwert 32). Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Nicht druckbare Zeichen löschen:** Löscht alle Steuerzeichen in der Zeichenfolge. Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Sonderzeichen dekodieren:** Sonderzeichen (oder Steuercodes) sind Zeichen, die keine Entsprechung auf der Tastatur haben, z. B. Wagenrücklauf oder Zeilenvorschub. verwendet eine menschenlesbare Codierung für solche Zeichen, beispielsweise <CR> für Wagenrücklauf und <LF> für Zeilenvorschub. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen (Steuercodes).
  Diese Option konvertiert Sonderzeichen aus der Syntax in tatsächliche Binärzeichen.

Beispiel

Wenn Sie die Datenfolge „<CR><LF>“ empfangen, fasst sie als reine Zeichenfolge aus 8 Zeichen auf. Um die empfangenen Daten als zwei Binärzeichen zu erkennen, **CR** (Wagenrücklauf – ASCII-Code 13) und **LF** (Zeilenvorschub – ASCII-Code 10), müssen Sie diese Option aktivieren.

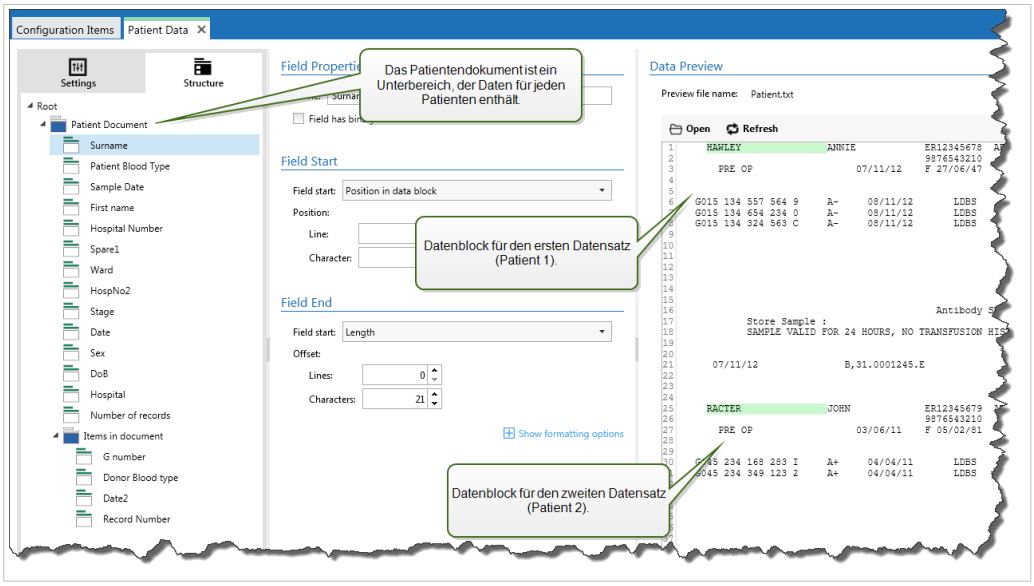
- **Suchen und Löschen von allem vor:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab Beginn der Daten bis zu der Zeichenfolge. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab der Zeichenfolge bis zum Ende der Daten. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Großschreibung ändern**: Ändert alle Zeichen in Ihren Zeichenfolgen zu Groß- oder Kleinschreibung.

2.2.3. Unterbereiche definieren

Ein Unterbereich ist ein Datenabschnitt, der mehrere Datenblöcke enthält, die von derselben Extraktionsregel erkannt werden. Jeder Datenblock stellt Daten für ein einzelnes Etikett bereit.

Alle Datenblöcke müssen mithilfe derselben Konfigurationsregel erkannt werden. Jeder Datenblock kann einen weiteren Unterbereich enthalten. Sie können eine unbegrenzte Anzahl von Unterbereichen in übergeordneten Unterbereichen definieren.

Beinhaltet der Filter die Definition eines Unterbereichs, zeigt die Aktion Datenfilter verwenden Unterbereiche mit eingebetteten Platzhaltern an. Alle Aktionen, die unter einem solchen Platzhalter eingebunden sind, werden nur für Datenblöcke auf dieser Ebene ausgeführt. Sie können verschiedene Etiketten mit Daten aus verschiedenen Unterbereichen drucken.



Konfiguration von Unterbereichen

Unterbereiche werden mit ähnlichen Regeln konfiguriert wie einzelne Felder. Jeder Unterbereich wird über die folgenden Parameter definiert.

- **Name Unterbereich:** Gibt den Namen des Unterbereichs an.
- **Datenblöcke:** Gibt an, wie Datenblöcke innerhalb des Unterbereichs erkannt werden. Jeder Unterbereich enthält mindestens einen Datenblock. Jeder Datenblock stellt Daten für ein einzelnes Etikett bereit.
    - **Jeder Block enthält eine feste Anzahl von Zeilen:** Gibt an, dass jeder Datenblock im Unterbereich exakt die eingegebene Anzahl von Zeilen enthält. Verwenden Sie diese Option, wenn Sie wissen, dass jeder Datenblock genau dieselbe Anzahl von Zeilen enthält.
    - **Blöcke beginnen mit einer Zeichenfolge:** Gibt an, dass Datenblöcke mit der eingegebenen Zeichenfolge beginnen. Alle Inhalte zwischen den beiden angegebenen Zeichenfolgen gehören zu einem separaten Datenblock. Der Inhalt zwischen der letzten Zeichenfolge und dem Ende der Daten stellt den letzten Datenblock dar.
    - **Blöcke enden mit einer Zeichenfolge:** Gibt an, dass Datenblöcke mit der eingegebenen Zeichenfolge enden. Alle Inhalte zwischen den beiden angegebenen Zeichenfolgen gehören zu einem separaten Datenblock. Der Inhalt zwischen dem Beginn der Daten und der ersten Zeichenfolge stellt den ersten Datenblock dar.
    - **Blöcke werden durch eine Zeichenfolge getrennt:** Gibt an, dass Datenblöcke durch die ausgewählte Zeichenfolge getrennt werden. Alle Inhalte zwischen den beiden ausgewählten Zeichenfolgen gehören zu einem separaten Datenblock.
- **Anfang des ersten Datenblocks:** Gibt die Anfangsposition des ersten Datenblocks an. Gleichzeitig definiert die Option die Anfangsposition des Unterbereichs. Für gewöhnlich ist die Anfangsposition auch der Anfang der empfangenen Daten. Die Konfigurationsparameter sind mit denen für die Definition von Feldern identisch. Weitere Informationen finden Sie im Abschnitt Felder Definieren.
- **Ende des letzten Datenblocks:** Gibt die Endposition des letzten Datenblocks an. Gleichzeitig definiert die Option die Endposition des Unterbereichs. Für gewöhnlich ist die Endposition das Ende der empfangenen Daten. Die Konfigurationsparameter sind mit denen für die Definition von Feldern identisch. Weitere Informationen finden Sie im Abschnitt Felder Definieren.

Konfiguration von Feldern innerhalb von Unterbereichen

Felder innerhalb des Unterbereichs werden anhand derselben Parameter konfiguriert wie auf Stammebene definierte Felder. Weitere Informationen finden Sie im Abschnitt Felder Definieren.

ANMERKUNG

Die Feld-Zeilennummern verweisen auf die Position innerhalb des Datenblocks, nicht auf die Position innerhalb der Eingangsdaten.

Datenvorschau
Dieser Bereich zeigt eine Vorschau der Felddefinition an. Wenn das definierte Element ausgewählt wird, zeigt die Vorschau dessen Position in den Vorschaudaten an.

- **Name der Vorschaudatei:** Gibt die Datei an, welche die Beispieldaten enthält, die durch den Filter geparst werden sollen. Die Vorschaudatei wird aus der Filterdefinition kopiert. Wenn Sie den Namen der Vorschaudatei ändern, wird der neue Name gespeichert.
- **Öffnen:** Wählt eine andere Datei aus, auf die die Filterregeln angewandt werden sollen.
- **Aktualisieren:** Wendet die Filterregeln erneut auf den Inhalt der Vorschaudatei an.Automation Aktualisiert den „Datenvorschau“-Bereich mit dem Ergebnis.

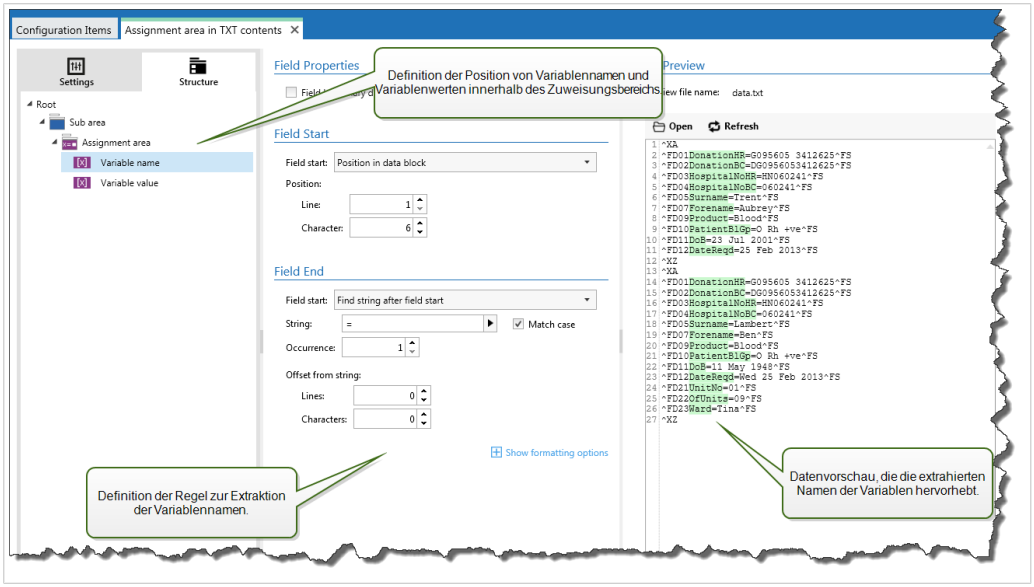
2.2.4. Zuweisungsbereiche definieren

Der Filter für unstrukturierte Daten erkennt Felder und ihre Werte in den empfangenen Daten automatisch. Dadurch ist eine manuelle *Variable-zu-Feld*-Zuordnung überflüssig.

Die Funktion **Dynamische Struktur** ist nützlich, wenn der Trigger Daten mit sich ändernder Struktur empfängt. In solchen Fällen bleibt die Haupt-Datenstruktur unverändert (z. B. Felder werden durch Kommas getrennt) oder dieselbe Struktur wird beibehalten, aber **die Reihenfolge** und/oder **die Anzahl** von Feldern ändert sich. Es könnten neue Felder hinzukommen oder alte Felder könnten nicht mehr verfügbar sein. Aufgrund der aktivierten **Dynamischen Struktur** erkennt der Filter die Struktur der empfangenen Datei automatisch. Gleichzeitig liest der Filter Feldnamen und Werte (`Name-Wert`-Paare) aus den Daten. Dadurch müssen Felder nicht manuell Variablen zugeordnet werden.

Die Aktion Datenfilter verwenden bietet keine Zuordnungsmöglichkeiten, da sie die Zuordnung dynamisch vornimmt. Sie müssen nicht einmal die Etikettenvariablen in der Trigger-Konfiguration festlegen. Die Aktion ordnet Feldwerte den gleichnamigen Etikettenvariablen zu, ohne dass die Variablen aus dem Etikett importiert werden müssen. Diese Regel gilt jedoch nur für die Aktion Etikett drucken. Wenn Sie die Feldwerte in einer anderen Aktion verwenden möchten, müssen Sie Variablen im Trigger definieren, dabei aber die automatische Zuordnung von *Variablen zu Feldern* beibehalten.

ANMERKUNG
Wenn ein Feld in den Eingabedaten keine entsprechende Variable auf dem Etikett hat, tritt kein Fehler auf. Ignoriert die fehlenden Variablen, ohne eine Meldung auszugeben.



Zuweisungsbereiche konfigurieren

Zuweisungsbereiche werden anhand derselben Methode konfiguriert wie Unterbereiche. Weitere Informationen finden Sie im Abschnitt Unterbereiche definieren. Der Zuweisungsbereich kann auf Stammdatenebene definiert werden, sodass er nur einmal vorhanden ist. Alternativ kann er auch innerhalb eines Unterbereichs konfiguriert werden, woraufhin er für jeden Datenblock im Unterbereich ausgeführt wird.

Felder im Zuweisungsbereich konfigurieren

Wenn Sie einen Zuweisungsbereich erstellen, definiert der Filter automatisch zwei Platzhalter. Diese zwei Platzhalter definieren das **`Name:Wert`**-Paar.

- **Variablenname:** Gibt das Feld an, dessen Inhalt als Variablenname verwendet wird (**`Name`**-Komponente in einem Paar). Konfigurieren Sie das Feld mithilfe derselben Methode, die Sie auch für Dokumentenfelder verwenden. Weitere Informationen finden Sie im Abschnitt Felder Definieren.
- **Variablenwert:** Gibt das Feld an, dessen Inhalt als Variablenwert verwendet wird (**`Wert`**-Komponente in einem Paar). Konfigurieren Sie das Feld mithilfe derselben Methode, die Sie auch für Dokumentenfelder verwenden. Weitere Informationen finden Sie im Abschnitt Felder Definieren.

Beispiel

Der Bereich zwischen ^XA und ^XZ ist der Zuweisungsbereich. Jede Zeile in einem Zuweisungsbereich stellt ein **`Name:Wert`**-Paar bereit. **Name** ist als der Wert zwischen dem 6. Zeichen und dem Gleichheitszeichen in jeder Zeile definiert. **Wert** ist als der Wert zwischen dem Gleichheitszeichen und dem Zeilenende (mit negativem Versatz um drei Zeichen) definiert.

```
^XA
^FD01DonationHR=G095605 3412625^FS
^FD02DonationBC=DG0956053412625^FS
^FD03HospitalNoHR=HN060241^FS
^FD04HospitalNoBC=060241^FS
^FD05Surname=Hawley^FS
^FD07Forename=Annie^FS
^FD09Product=Blood^FS
^FD10PatientBlGp=O Rh +ve^FS
^FD11DoB=27 June 1947^FS
^FD12DateReqd=25 Dec 2012^FS
^XZ
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

2.3. XML-Filter konfigurieren

2.3.1. XML-Filter

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Um mehr über Filter im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Filtern.

Verwenden Sie diesen Filter, wenn ein Trigger XML-codierte Daten erhält. Der Filter ermöglicht es Ihnen, einzelne Felder, Felder in sich wiederholenden Unterbereichen und sogar **`Name-Wert`**-Paare zu extrahieren. Die XML-Struktur gibt Elemente und Unterelemente, Attribute und ihre Werte sowie Textwerte (Elementwerte) vor.

Obwohl Sie die Struktur der XML-Datei selbst definieren können, empfiehlt NiceLabel Ihnen, die Struktur aus der vorhandenen XML-Beispieldatei zu importieren. Klicken Sie auf **Datenstruktur importieren** in der Menüleiste. Nachdem Sie die XML-Struktur importiert haben, wird im Abschnitt, Datenvorschau“ der XML-Inhalt angezeigt. Außerdem hebt dieser Abschnitt die Elemente und Attribute hervor, die Sie als Ausgabefelder definieren.

Beispiele für XML-Daten finden Sie im Abschnitt XML-Daten.

Struktur definieren
Um die XML-Objekte zu verwenden, konfigurieren Sie sie wie folgt:

- **Variablenwert:** Gibt an, dass Sie das ausgewählte Element als Feld verwenden möchten und seinen Wert den entsprechenden Variablen in der Aktion Datenfilter verwenden zuordnen werden. Weitere Informationen finden Sie im Abschnitt XML-Felder definieren.
    - **Optionales Element:** Gibt an, dass das Element nicht obligatorisch ist. Dies entspricht dem XML-Schema-(XSD-Datei-)Attribut `minOccurs=0`. Die einem solchen Feld zugeordnete Variable hat einen leeren Wert, wenn das Element nicht im XML erscheint.
- **Datenblöcke:** Gibt an, dass das ausgewählte Element mehrere Male vorkommt und Daten für ein einzelnes Etikett bereitstellt. Ein Datenblock kann als sich wiederholender Bereich, als Zuweisungsbereich oder beides definiert werden.
    - **Wiederholbarer Bereich:** Gibt an, dass Sie Werte aus allen wiederholt auftretenden Datenblöcken extrahieren wollen, nicht nur aus dem ersten. Sie können Felder innerhalb jedes Datenblocks definieren. Ordnen Sie in der Aktion Datenfilter verwenden die definierten Felder den jeweiligen Variablen zu. Weitere Informationen finden Sie im Abschnitt Wiederholbare Elemente Definieren.
    - **Zuweisungsbereich:** Gibt an, dass der Datenblock `Name-Wert` Paare enthält. Feldnamen und ihre Werte werden gleichzeitig ausgelesen. Die Zuordnung zu Variablen erfolgt automatisch. Verwenden Sie diese Funktion, um den Filter für wechselnde Eingangsdaten einzurichten; so können Sie Wartungsaufwand vermeiden. Weitere Informationen finden Sie im Abschnitt XML-Zuweisungsbereich definieren.

Der Bereich „Datenvorschau“ vereinfacht die Konfiguration. Das Ergebnis der definierten Filterregeln wird im Vorschaubereich hervorgehoben.

Um die als Vorschau angezeigten XML-Daten zu ändern klicken Sie auf **Öffnen** und suchen Sie nach einer neuen Beispiel-XML-Datei.

2.3.2. XML-Felder definieren

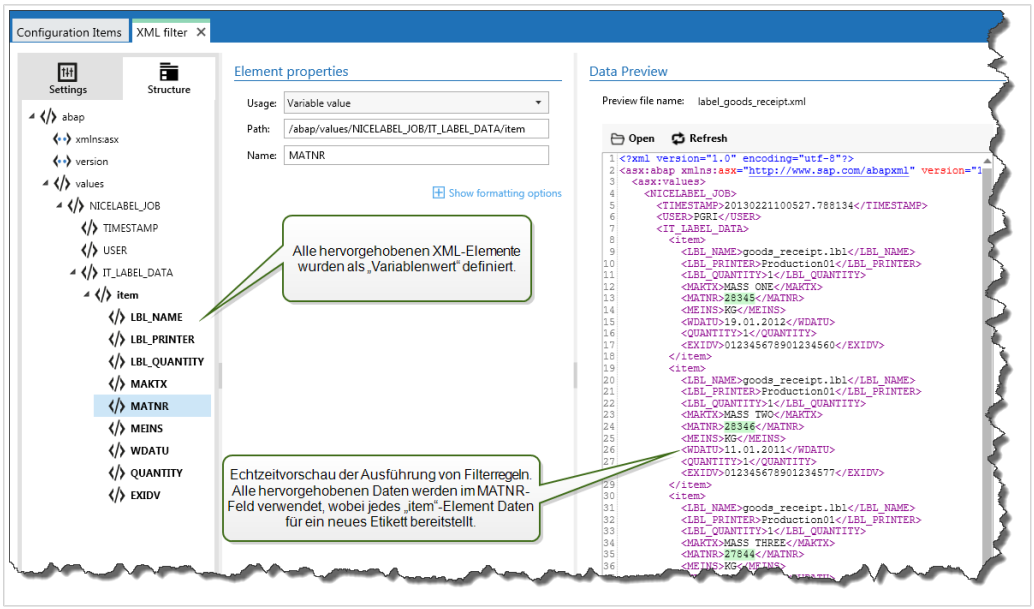
PRODUKTEBENEN-INFO
Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Beim Definieren von XML-Feldern stellen Sie Werte der ausgewählten Elemente automatisch zur Verfügung. Die Filterdefinition stellt solche Felder für die Zuordnung zu Variablen in Aktionen zur Verfügung. Auf diese Weise können Sie Werte von Elementen oder Attributen extrahieren.

So definieren Sie einen Elementwert als Feld:

1. Wählen Sie das Element oder Attribut in der Strukturliste aus.

2. Wählen Sie unter **Nutzung** die Option **Variablenwert** aus.
3. Das Element wird in der Strukturliste in Fettdruck angezeigt, um zu verdeutlichen, dass es verwendet wird.
4. Der Element- oder Attributname wird als Name des Ausgabefeldes verwendet.
5. Der Abschnitt „Datenvorschau“ hebt den Wert des ausgewählten Elements hervor.



Formatierungsoptionen

Dieser Abschnitt definiert Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden in der Reihenfolge ausgeführt, die in der Benutzeroberfläche ausgewählt ist – von oben nach unten.

- **Leerzeichen am Anfang löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Anfang einer Zeichenfolge.
- **Leerzeichen am Ende löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Ende einer Zeichenfolge.
- **Eröffnungs- und Abschlusszeichen löschen:** Löscht die erste Instanz der ausgewählten Eröffnungs- und Abschlusszeichen, die in einer Zeichenfolge enthalten sind.

Beispiel

Wenn Sie „{“ als Eröffnungszeichen und „}“ als Abschlusszeichen verwenden, wird die Eingabe-Zeichenfolge **`{{Auswahl}}`** in **`{Auswahl}`** konvertiert.

- **Suchen und ersetzen:** Führt anhand der angegebenen Werte für *Finde* und *Ersetzen durch* eine Standardfunktion für Suchen und Ersetzen durch. Sie können auch reguläre Ausdrücke verwenden.

  ANMERKUNG

  Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. Nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im Knowledge Base-Artikel.

- **Nicht-druckbare Zeichen durch Leerzeichen ersetzen:** Ersetzt alle Steuerzeichen in einer Zeichenfolge durch Leerzeichen (ASCII-Dezimalwert 32). Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Nicht druckbare Zeichen löschen:** Löscht alle Steuerzeichen in der Zeichenfolge. Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Sonderzeichen dekodieren:** Sonderzeichen (oder Steuercodes) sind Zeichen, die keine Entsprechung auf der Tastatur haben, z. B. Wagenrücklauf oder Zeilenvorschub. verwendet eine menschenlesbare Codierung für solche Zeichen, beispielsweise <CR> für Wagenrücklauf und <LF> für Zeilenvorschub. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen (Steuercodes).
  Diese Option konvertiert Sonderzeichen aus der Syntax in tatsächliche Binärzeichen.

Beispiel

Wenn Sie die Datenfolge „<CR><LF>“ empfangen, fasst sie als reine Zeichenfolge aus 8 Zeichen auf. Um die empfangenen Daten als zwei Binärzeichen zu erkennen, **CR** (Wagenrücklauf – ASCII-Code 13) und **LF** (Zeilenvorschub – ASCII-Code 10), müssen Sie diese Option aktivieren.

- **Suchen und Löschen von allem vor:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab Beginn der Daten bis zu der Zeichenfolge. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab der Zeichenfolge bis zum Ende der Daten. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Großschreibung ändern**: Ändert alle Zeichen in Ihren Zeichenfolgen zu Groß- oder Kleinschreibung.

Datenvorschau
Dieser Bereich zeigt eine Vorschau der Felddefinition an. Wenn das definierte Element ausgewählt wird, zeigt die Vorschau dessen Position in den Vorschaudaten an.

- **Name der Vorschaudatei:** Gibt die Datei an, welche die Beispieldaten enthält, die durch den Filter geparst werden sollen. Die Vorschaudatei wird aus der Filterdefinition kopiert. Wenn Sie den Namen der Vorschaudatei ändern, wird der neue Name gespeichert.
- **Öffnen:** Wählt eine andere Datei aus, auf die die Filterregeln angewandt werden sollen.
- **Aktualisieren:** Wendet die Filterregeln erneut auf den Inhalt der Vorschaudatei an.Automation Aktualisiert den „Datenvorschau“-Bereich mit dem Ergebnis.

2.3.3. Wiederholbare Elemente im XML-Filter definieren

PRODUKTEBENEN-INFO
Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Wenn ein XML-Element mehrmals in XML-Daten vorkommt, handelt es sich um ein sich wiederholendes Element. Für gewöhnlich enthalten solche Elemente Daten für ein einzelnes Etikett. Um anzugeben, dass Sie Daten aus allen wiederholt auftretenden Elementen verwenden wollen, und nicht nur aus dem ersten, definieren Sie das Element als **Datenblock** und aktivieren Sie die Option **Wiederholbares Element**. Beinhaltet der Filter die Definition von Elementen, die als Datenblock / wiederholbare Elemente definiert sind, zeigt die Aktion Datenfilter verwenden wiederholbare Elemente mit eingebetteten Platzhaltern an. Alle Aktionen, die unter einem solchen Platzhalter eingebunden sind, werden nur für Datenblöcke auf dieser Ebene ausgeführt.

Beispiel

Das Element **<item>** ist als **Datenblock** und als **Wiederholbares Element** definiert. Dadurch erhält der Filter die Anweisung, alle Instanzen des Elements **<item>** zu extrahieren, nicht nur die erste. In diesem Fall sollte **<item>** als Unterebene in der Aktion **Datenfilter verwenden** definiert werden. Sie müssen die Aktionen „Etikett öffnen“ und „Etikett drucken“ unter diesem Unterebenen-Platzhalter einbetten, damit sie für jede Instanz des Elements **<item>** wiederholt werden. Im folgenden Beispiel geschieht dies für drei Instanzen.

```
<?xml ver sion="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" ver sion="1.0">
 <asx:values>
  <NICELABEL_JOB>
   <T IMEST AMP>20130221100527.788134</T IMEST AMP>
   <USER>PGRI</USER>
   <IT _LABEL_DAT A>
   <item>
      <LBL_NAME>goods_r eceipt.nlbl</LBL_NAME>
      <LBL_PRINT ER>Pr oduction01</LBL_PRINT ER>
      <LBL_QUANT IT Y>1</LBL_QUANT IT Y>
      <MAKT X>MASS ONE</MAKT X>
      <MAT NR>28345</MAT NR>
      <MEINS>KG</MEINS>
      <WDAT U>19.01.2012</WDAT U>
      <QUANT IT Y>1</QUANT IT Y>
      <EXIDV>012345678901234560</EXIDV>
   </item>

   <item>
      <LBL_NAME>goods_r eceipt.nlbl</LBL_NAME>
      <LBL_PRINT ER>Pr oduction01</LBL_PRINT ER>
      <LBL_QUANT IT Y>1</LBL_QUANT IT Y>
      <MAKT X>MASS T WO</MAKT X>
      <MAT NR>28346</MAT NR>
      <MEINS>KG</MEINS>
      <WDAT U>11.01.2011</WDAT U>
      <QUANT IT Y>1</QUANT IT Y>
      <EXIDV>012345678901234577</EXIDV>
   </item>

   <item>
      <LBL_NAME>goods_r eceipt.nlbl</LBL_NAME>
      <LBL_PRINT ER>Pr oduction01</LBL_PRINT ER>
      <LBL_QUANT IT Y>1</LBL_QUANT IT Y>
      <MAKT X>MASS T HREE</MAKT X>
      <MAT NR>27844</MAT NR>
      <MEINS>KG</MEINS>
```

```
      <WDAT U>07.03.2009</WDAT U>
      <QUANT IT Y>1</QUANT IT Y>
      <EXIDV>012345678901234584</EXIDV>
   </item>

  </IT _LABEL_DAT A>
 </NICELABEL_JOB>
 </asx:values>
</asx:abap>
```

2.3.4. XML-Zuweisungsbereich definieren

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

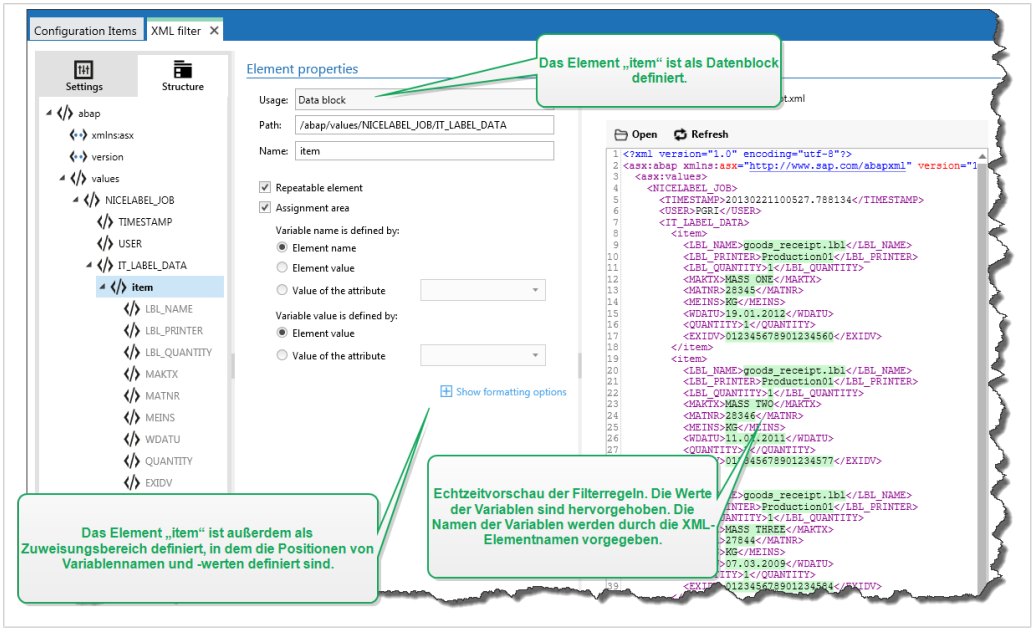
Der XML-Filter erkennt Felder und ihre Werte in den empfangenen Daten automatisch. Daher ist eine manuelle *Variable-zu-Feld*-Zuordnung nicht notwendig.

Die Funktion **Dynamische Struktur** ist nützlich, wenn der Trigger Daten mit sich ändernder Struktur empfängt. In solchen Fällen bleibt die Haupt-Datenstruktur unverändert (z. B. Felder werden durch Kommas getrennt) oder dieselbe Struktur wird beibehalten, aber **die Reihenfolge** und/oder **die Anzahl** von Feldern ändert sich. Es könnten neue Felder hinzukommen oder alte Felder könnten nicht mehr verfügbar sein. Aufgrund der aktivierten **Dynamischen Struktur** erkennt der Filter die Struktur der empfangenen Datei automatisch. Gleichzeitig liest der Filter Feldnamen und Werte (**`Name-Wert`**-Paare) aus den Daten. Dadurch müssen Felder nicht manuell Variablen zugeordnet werden.

Die Aktion Datenfilter verwenden bietet keine Zuordnungsmöglichkeiten, da sie die Zuordnung dynamisch vornimmt. Sie müssen nicht einmal die Etikettenvariablen in der Trigger-Konfiguration festlegen. Die Aktion ordnet Feldwerte den gleichnamigen Etikettenvariablen zu, ohne dass die Variablen aus dem Etikett importiert werden müssen. Diese Regel gilt jedoch nur für die Aktion Etikett drucken. Wenn Sie die Feldwerte in einer anderen Aktion verwenden möchten, müssen Sie Variablen im Trigger definieren, dabei aber die automatische Zuordnung von *Variablen zu Feldern* beibehalten.

ANMERKUNG

Wenn ein Feld in den Eingabedaten keine entsprechende Variable auf dem Etikett hat, tritt kein Fehler auf. Ignoriert die fehlenden Variablen, ohne eine Meldung auszugeben.



XML-Zuweisungsbereiche konfigurieren

Wenn Sie den Datenblock als Zuweisungsbereich definieren, werden unter der Definition dieses Elements zwei Platzhalter angezeigt. Sie müssen angeben, wie der Feldname und -wert definiert sind, damit der Filter das **`Name-Wert`**-Paar extrahieren kann.

- **Variablenname:** Gibt das Element an, das den Feldnamen enthält. Der Name kann nach Elementname, ausgewähltem Attributwert oder Elementwert definiert werden. Um automatische Zuordnung zu aktivieren, muss die Etikettenvariable denselben Namen haben.
- **Variablenwert:** Gibt das Element an, das den Feldwert enthält. Der Name kann nach dem Elementnamen, ausgewählten Attributwert oder Elementwert definiert werden.

WARNUNG

Das XML-Element, das **`Name:Wert`** Paare enthält, darf nicht als Stammelement festgelegt werden, muss sich aber mindestens auf der zweiten Ebene befinden. Im folgenden XML-Beispiel befindet sich das Element **`<label>`** z. B. auf der zweiten Ebene und kann die **`Name:Wert`** Paare enthalten.

Formatierungsoptionen

Dieser Abschnitt definiert Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden in der Reihenfolge ausgeführt, die in der Benutzeroberfläche ausgewählt ist – von oben nach unten.

- **Leerzeichen am Anfang löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Anfang einer Zeichenfolge.
- **Leerzeichen am Ende löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Ende einer Zeichenfolge.
- **Eröffnungs- und Abschlusszeichen löschen:** Löscht die erste Instanz der ausgewählten Eröffnungs- und Abschlusszeichen, die in einer Zeichenfolge enthalten sind.

Beispiel

Wenn Sie „{“ als Eröffnungszeichen und „}“ als Abschlusszeichen verwenden, wird die Eingabe-Zeichenfolge **`{{Auswahl}}`** in **`{Auswahl}`** konvertiert.

- **Suchen und ersetzen:** Führt anhand der angegebenen Werte für *Finde* und *Ersetzen durch* eine Standardfunktion für Suchen und Ersetzen durch. Sie können auch reguläre Ausdrücke verwenden.

  ANMERKUNG

  Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. Nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im Knowledge Base-Artikel.

- **Nicht-druckbare Zeichen durch Leerzeichen ersetzen:** Ersetzt alle Steuerzeichen in einer Zeichenfolge durch Leerzeichen (ASCII-Dezimalwert 32). Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Nicht druckbare Zeichen löschen:** Löscht alle Steuerzeichen in der Zeichenfolge. Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Sonderzeichen dekodieren:** Sonderzeichen (oder Steuercodes) sind Zeichen, die keine Entsprechung auf der Tastatur haben, z. B. Wagenrücklauf oder Zeilenvorschub. verwendet eine menschenlesbare Codierung für solche Zeichen, beispielsweise <CR> für Wagenrücklauf und <LF> für Zeilenvorschub. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen (Steuercodes).
  Diese Option konvertiert Sonderzeichen aus der Syntax in tatsächliche Binärzeichen.

Beispiel

Wenn Sie die Datenfolge „<CR><LF>“ empfangen, fasst sie als reine Zeichenfolge aus 8 Zeichen auf. Um die empfangenen Daten als zwei Binärzeichen zu erkennen, **CR** (Wagenrücklauf – ASCII-Code 13) und **LF** (Zeilenvorschub – ASCII-Code 10), müssen Sie diese Option aktivieren.

- **Suchen und Löschen von allem vor:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab Beginn der Daten bis zu der Zeichenfolge. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab der Zeichenfolge bis zum Ende der Daten. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Großschreibung ändern**: Ändert alle Zeichen in Ihren Zeichenfolgen zu Groß- oder Kleinschreibung.

Beispiel

Das Element **<label>** ist als Datenblock und als Zuweisungsbereich definiert. Der **Variablenname** wird anhand des Wertes des Attributnamens, **der Variablenwert** anhand des Elementtexts bereitgestellt.

```
<?xml version="1.0" standalone="no"?>
<labels _FORMAT="case.nlbl" _PRINTERNAME="Production01" _QUANTITY="1">
    <label>
        <variable name="CASEID">0000000123</variable>
        <variable name="CARTONTYPE"/>
        <variable name="ORDERKEY">0000000534</variable>
        <variable name="BUYERPO"/>
        <variable name="ROUTE"> </variable>
        <variable name="CONTAINERDETAILID">0000004212</variable>
        <variable name="SERIALREFERENCE">0</variable>
        <variable name="FILTERVALUE">0</variable>
        <variable name="INDICATORDIGIT">0</variable>
        <variable name="DATE">11/19/2012 10:59:03</variable>
    </label>
</labels>
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

2.4. JSON-Filter konfigurieren

2.4.1. JSON-Filter

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Um mehr über Filter im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Filtern.

Verwenden Sie den JSON-Filter, wenn Ihr Trigger JSON-codierte Daten erhält. Mit dem JSON-Filter können Sie Veriablen und Werte aus Ihrer JSON-Datei verwenden. Der Filter unterstützt die Datenextraktion aus JSON-Datenfeldern.

ANMERKUNG

Automation ermöglicht Ihnen die Nutzung aller JSON-Datentypen. Weitere Informationen über die verfügbaren JSON-Datentypen finden Sie hier.

Obwohl Sie die JSON-Dateistruktur manuell definieren können, empfiehlt NiceLabel Ihnen, die Struktur aus den empfangenen JSON-Dateien zu importieren.

So importieren Sie die JSON-Dateistruktur:

1. Gehen Sie auf **Datenfilter** und **Bearbeiten** Sie Ihren JSON-Filter.
2. Klicken Sie auf **Struktur** > **Datenstruktur importieren**. Navigieren Sie zu Ihrer JSON-Datei und klicken Sie auf **Öffnen**.
   Nach dem Importieren von JSON-Dateien werden die JSON-Inhalte im Bereich **Datenvorschau** angezeigt. Außerdem hebt die Datenvorschau die Elemente hervor, die Sie als Ausgabefelder definieren.

Beispiele für JSON-Daten finden Sie im Abschnitt JSON-Daten.

Struktur definieren
Um die JSON-Objekte zu verwenden, konfigurieren Sie sie wie folgt:

- **Variablenwert:** Legt fest, dass Sie das ausgewählte Objekt als Feld verwenden möchten. Beim Erstellen der Konfiguration ordnen Sie Werte den entsprechenden Variablen in der Aktion Datenfilter verwenden zu. Weitere Informationen finden Sie im Abschnitt JSON-Felder definieren.
    - **Optionales Element**: Gibt an, dass das Element nicht obligatorisch ist. Die einem solchen Feld zugeordnete Variable hat einen leeren Wert, wenn das Element nicht in der JSON-Datei vorkommt.
- **Datenblöcke:** Gibt an, dass die enthaltenen Unterelemente mehrere Male vorhanden sind und Daten für Ihre Etiketten bereitstellen. Ein Datenblock kann als sich wiederholender Bereich, als Zuweisungsbereich oder beides definiert werden. Im Fall von JSON fungiert der Datenblock als Datenfeld.
    - **Wiederholbarer Bereich:** Gibt an, dass Sie Werte aus allen wiederholt auftretenden Datenblöcken extrahieren wollen, nicht nur aus dem ersten. Sie können Felder innerhalb jedes Datenblocks definieren. Ordnen Sie in der Aktion Datenfilter verwenden manuell die definierten Felder den jeweiligen Variablen zu. Weitere Informationen finden Sie im Abschnitt Wiederholbare Elemente im JSON-Filter definieren.
    - **Zuweisungsbereich:** Erstellt automatisch die Variablen und ordnet sie relevanten Werten zu. Feldnamen und ihre Werte werden gleichzeitig ausgelesen. Die Zuordnung zu Variablen erfolgt automatisch. Verwenden Sie diese Funktion, um den Filter für wechselnde Eingangsdaten einzurichten; so können Sie Wartungsaufwand vermeiden. Weitere Informationen finden Sie im Abschnitt JSON-Zuweisungsbereich definieren.

Der Bereich **Datenvorschau** vereinfacht die Konfiguration. Das Ergebnis der definierten Filterregeln wird im Vorschaubereich hervorgehoben.

Um die als Vorschau angezeigten JSON-Daten zu ändern klicken Sie auf **Öffnen** und suchen Sie nach einer neuen Beispiel-JSON-Datei.

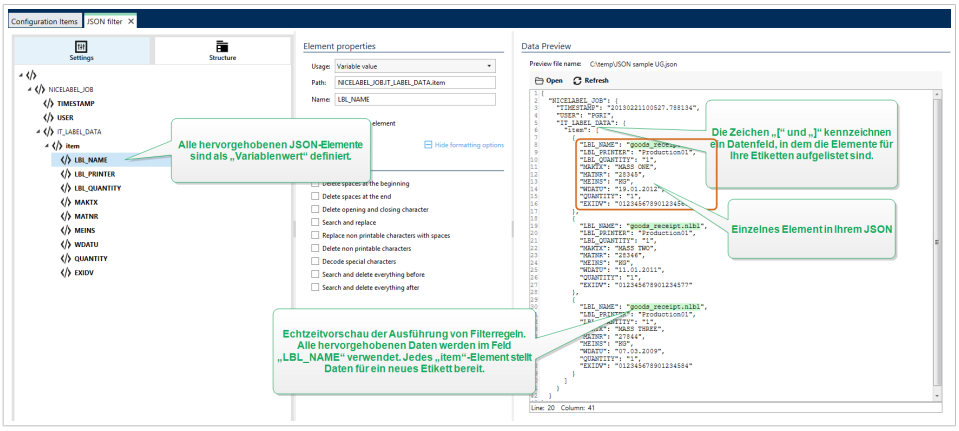
2.4.2. JSON-Felder definieren

PRODUKTEBENEN-INFO
Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Beim Definieren von JSON-Feldern stellen Sie Werte der ausgewählten Elemente automatisch zur Verfügung. Ihre Filterdefinition stellt solche Felder für die Zuordnung zu Variablen in Aktionen zur Verfügung. Auf diese Weise können Sie die Elementwerte extrahieren.

So definieren Sie JSON-Felder:

1. Wählen Sie Ihr Element aus und stellen Sie seine **Nutzung** auf **Variabler Wert** ein.
2. Das Element wird in der Strukturliste in Fettdruck angezeigt, um zu verdeutlichen, dass es verwendet wird.
3. Der Element wird als Name des Ausgabefeldes verwendet.
4. Der Abschnitt „Datenvorschau“ hebt die Werte des ausgewählten Elements hervor.



Formatierungsoptionen

Dieser Abschnitt definiert Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden in der Reihenfolge ausgeführt, die in der Benutzeroberfläche ausgewählt ist – von oben nach unten.

- **Leerzeichen am Anfang löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Anfang einer Zeichenfolge.
- **Leerzeichen am Ende löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Ende einer Zeichenfolge.
- **Eröffnungs- und Abschlusszeichen löschen:** Löscht die erste Instanz der ausgewählten Eröffnungs- und Abschlusszeichen, die in einer Zeichenfolge enthalten sind.

Beispiel

Wenn Sie „{“ als Eröffnungszeichen und „}“ als Abschlusszeichen verwenden, wird die Eingabe-Zeichenfolge **`{{Auswahl}}`** in **`{Auswahl}`** konvertiert.

- **Suchen und ersetzen:** Führt anhand der angegebenen Werte für *Finde* und *Ersetzen durch* eine Standardfunktion für Suchen und Ersetzen durch. Sie können auch reguläre Ausdrücke verwenden.

  ANMERKUNG

  Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. Nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im Knowledge Base-Artikel.

- **Nicht-druckbare Zeichen durch Leerzeichen ersetzen:** Ersetzt alle Steuerzeichen in einer Zeichenfolge durch Leerzeichen (ASCII-Dezimalwert 32). Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Nicht druckbare Zeichen löschen:** Löscht alle Steuerzeichen in der Zeichenfolge. Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Sonderzeichen dekodieren:** Sonderzeichen (oder Steuercodes) sind Zeichen, die keine Entsprechung auf der Tastatur haben, z. B. Wagenrücklauf oder Zeilenvorschub. verwendet eine menschenlesbare Codierung für solche Zeichen, beispielsweise <CR> für Wagenrücklauf und <LF> für Zeilenvorschub. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen (Steuercodes).
  Diese Option konvertiert Sonderzeichen aus der Syntax in tatsächliche Binärzeichen.

Beispiel

Wenn Sie die Datenfolge „<CR><LF>“ empfangen, fasst sie als reine Zeichenfolge aus 8 Zeichen auf. Um die empfangenen Daten als zwei Binärzeichen zu erkennen, **CR** (Wagenrücklauf – ASCII-Code 13) und **LF** (Zeilenvorschub – ASCII-Code 10), müssen Sie diese Option aktivieren.

- **Suchen und Löschen von allem vor:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab Beginn der Daten bis zu der Zeichenfolge. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab der Zeichenfolge bis zum Ende der Daten. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Großschreibung ändern**: Ändert alle Zeichen in Ihren Zeichenfolgen zu Groß- oder Kleinschreibung.

Datenvorschau
Dieser Bereich zeigt eine Vorschau der Felddefinition an. Wenn das definierte Element ausgewählt wird, zeigt die Vorschau dessen Position in den Vorschaudaten an.

- **Name der Vorschaudatei:** Gibt die Datei an, welche die Beispieldaten enthält, die durch den Filter geparst werden sollen. Die Vorschaudatei wird aus der Filterdefinition kopiert. Wenn Sie den Namen der Vorschaudatei ändern, wird der neue Name gespeichert.
- **Öffnen:** Wählt eine andere Datei aus, auf die die Filterregeln angewandt werden sollen.
- **Aktualisieren:** Wendet die Filterregeln erneut auf den Inhalt der Vorschaudatei an.Automation Aktualisiert den „Datenvorschau“-Bereich mit dem Ergebnis.

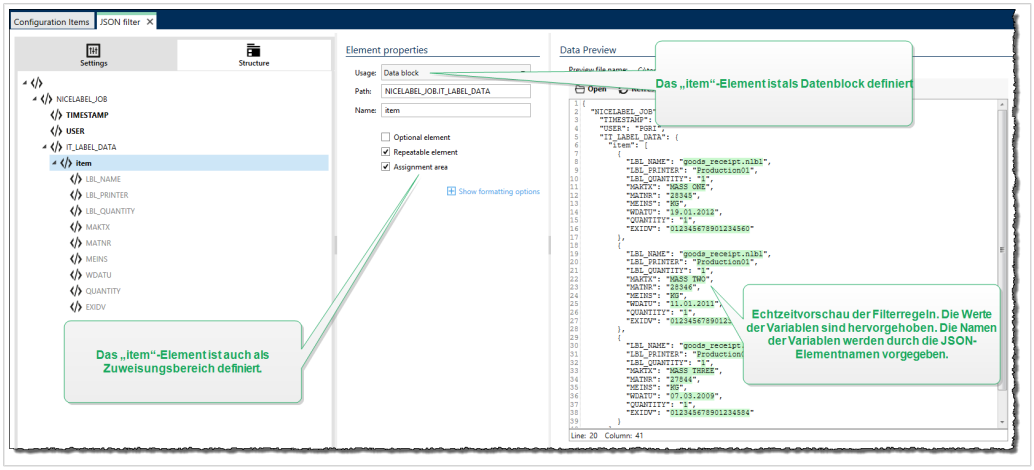
2.4.3. Wiederholbare Elemente im JSON-Filter definieren

PRODUKTEBENEN-INFO
Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Wenn ein JSON-Element mehrmals in Ihren JSON-Daten vorkommt, handelt es sich um ein wiederholbares Element. Für gewöhnlich enthalten wiederholbare Elemente Daten für ein einzelnes Etikett. Wiederholbare Elemente erzeugen mehrere Etiketten, die mit relevanten Daten ausgefüllt werden.

So geben Sie an, dass Sie Daten aus allen wiederholbaren Elementen extrahieren wollen, nicht nur aus dem ersten:

1. Wählen Sie das Element aus und definieren Sie es als **Datenblock**.
2. Aktivieren Sie die Option **Wiederholbares Element**.



Beinhaltet der Filter die Definition von Elementen, die als Datenblock / wiederholbare Elemente definiert sind, zeigt die Aktion Datenfilter verwenden wiederholbare Elemente mit eingebetteten Platzhaltern an. Alle Aktionen, die unter einem solchen Platzhalter eingebunden sind, werden nur für Datenblöcke auf dieser Ebene ausgeführt.

Beispiel

Das **„item“**-Element ist sowohl als **Datenblock** als auch als **Wiederholbares Element** definiert. Dadurch erhält der Filter die Anweisung, alle Instanzen des Datenfelds zu extrahieren, nicht nur die erste. In diesem Fall sollte **„item“** als Unterebene in der Aktion **Datenfilter verwenden** definiert werden. Sie müssen die Aktionen „Etikett öffnen“ und „Etikett drucken“ unter diesem Unterebenen-Platzhalter einbetten, damit sie für jede Instanz des **„item“**-Elements wiederholt werden. Im folgenden Beispiel geschieht dies für drei Instanzen.


```
{
    "NICELABEL_JOB": {
        "TIMESTAMP": "20130221100527.788134",
        "USER": "PGRI",
        "IT_LABEL_DATA": {
            "item": [
                {
                    "LBL_NAME": "goods_receipt.nlbl",
                    "LBL_PRINTER": "Production01",
                    "LBL_QUANTITY": "1",
                    "MAKTX": "MASS ONE",
                    "MATNR": "28345",
                    "MEINS": "KG",
                    "WDATU": "19.01.2012",
                    "QUANTITY": "1",
                    "EXIDV": "012345678901234560"
                },
                {
                    "LBL_NAME": "goods_receipt.nlbl",
                    "LBL_PRINTER": "Production01",
                    "LBL_QUANTITY": "1",
                    "MAKTX": "MASS TWO",
                    "MATNR": "28346",
                    "MEINS": "KG",
                    "WDATU": "11.01.2011",
                    "QUANTITY": "1",
                    "EXIDV": "012345678901234577"
                },
                {
                    "LBL_NAME": "goods_receipt.nlbl",
                    "LBL_PRINTER": "Production01",
                    "LBL_QUANTITY": "1",
                    "MAKTX": "MASS THREE",
                    "MATNR": "27844",
                    "MEINS": "KG",
                    "WDATU": "07.03.2009",
                    "QUANTITY": "1",
                    "EXIDV": "012345678901234584"
```

```
                }
            ]
        }
    }
}
```

2.4.4. JSON-Zuweisungsbereich definieren

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

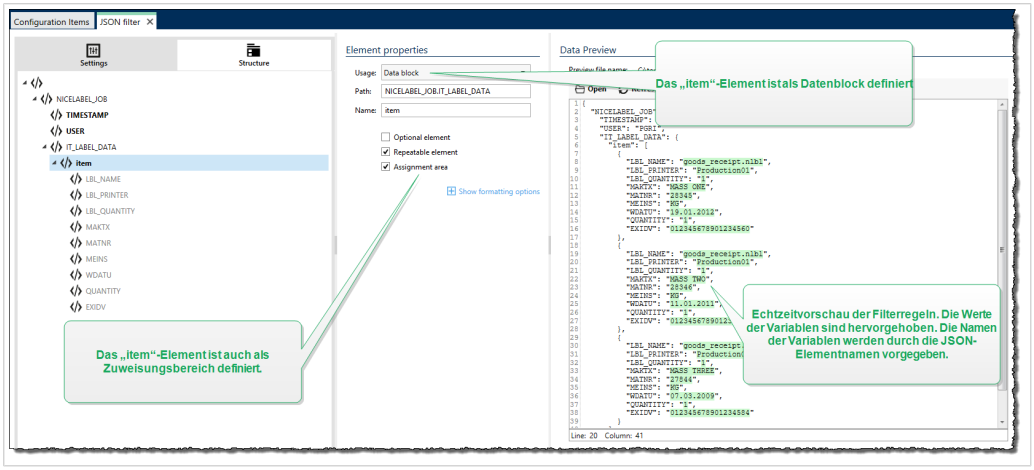
Der JSON-Filter erkennt Felder und ihre Werte in den empfangenen Daten automatisch. Daher ist eine manuelle *Variable-zu-Feld*-Zuordnung nicht notwendig.

Die Funktion **Dynamische Struktur** ist nützlich, wenn der Trigger Daten mit sich ändernder Struktur empfängt. In solchen Fällen bleibt die Haupt-Datenstruktur unverändert (z. B. Felder werden durch Kommas getrennt) oder dieselbe Struktur wird beibehalten, aber **die Reihenfolge** und/oder **die Anzahl** von Feldern ändert sich. Es könnten neue Felder hinzukommen oder alte Felder könnten nicht mehr verfügbar sein. Aufgrund der aktivierten **Dynamischen Struktur** erkennt der Filter die Struktur der empfangenen Datei automatisch. Gleichzeitig liest der Filter Feldnamen und Werte (**`Name-Wert`**-Paare) aus den Daten. Dadurch müssen Felder nicht manuell Variablen zugeordnet werden.

Die Aktion Datenfilter verwenden bietet keine Zuordnungsmöglichkeiten, da sie die Zuordnung dynamisch vornimmt. Sie müssen nicht einmal die Etikettenvariablen in der Trigger-Konfiguration festlegen. Die Aktion ordnet Feldwerte den gleichnamigen Etikettenvariablen zu, ohne dass die Variablen aus dem Etikett importiert werden müssen. Diese Regel gilt jedoch nur für die Aktion Etikett drucken. Wenn Sie die Feldwerte in einer anderen Aktion verwenden möchten, müssen Sie Variablen im Trigger definieren, dabei aber die automatische Zuordnung von *Variablen zu Feldern* beibehalten.

ANMERKUNG

Wenn ein Feld in den Eingabedaten keine entsprechende Variable auf dem Etikett hat, tritt kein Fehler auf. Ignoriert die fehlenden Variablen, ohne eine Meldung auszugeben.



ANMERKUNG

Da es in JSON keine optionalen Attribute gibt, definiert Automation **Variablennamen** und **Variable Werte** automatisch.

Formatierungsoptionen

Dieser Abschnitt definiert Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden in der Reihenfolge ausgeführt, die in der Benutzeroberfläche ausgewählt ist – von oben nach unten.

- **Leerzeichen am Anfang löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Anfang einer Zeichenfolge.
- **Leerzeichen am Ende löschen:** Löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Ende einer Zeichenfolge.
- **Eröffnungs- und Abschlusszeichen löschen:** Löscht die erste Instanz der ausgewählten Eröffnungs- und Abschlusszeichen, die in einer Zeichenfolge enthalten sind.

Beispiel

Wenn Sie „{“ als Eröffnungszeichen und „}“ als Abschlusszeichen verwenden, wird die Eingabe-Zeichenfolge **`{{Auswahl}}`** in **`{Auswahl}`** konvertiert.

- **Suchen und ersetzen:** Führt anhand der angegebenen Werte für *Finde* und *Ersetzen durch* eine Standardfunktion für Suchen und Ersetzen durch. Sie können auch reguläre Ausdrücke verwenden.

  ANMERKUNG

  Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. Nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im Knowledge Base-Artikel.

- **Nicht-druckbare Zeichen durch Leerzeichen ersetzen:** Ersetzt alle Steuerzeichen in einer Zeichenfolge durch Leerzeichen (ASCII-Dezimalwert 32). Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Nicht druckbare Zeichen löschen:** Löscht alle Steuerzeichen in der Zeichenfolge. Die nicht druckbaren Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Sonderzeichen dekodieren:** Sonderzeichen (oder Steuercodes) sind Zeichen, die keine Entsprechung auf der Tastatur haben, z. B. Wagenrücklauf oder Zeilenvorschub. verwendet eine menschenlesbare Codierung für solche Zeichen, beispielsweise <CR> für Wagenrücklauf und <LF> für Zeilenvorschub. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen (Steuercodes).
  Diese Option konvertiert Sonderzeichen aus der Syntax in tatsächliche Binärzeichen.

Beispiel

Wenn Sie die Datenfolge „<CR><LF>“ empfangen, fasst sie als reine Zeichenfolge aus 8 Zeichen auf. Um die empfangenen Daten als zwei Binärzeichen zu erkennen, **CR** (Wagenrücklauf – ASCII-Code 13) und **LF** (Zeilenvorschub – ASCII-Code 10), müssen Sie diese Option aktivieren.

- **Suchen und Löschen von allem vor:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab Beginn der Daten bis zu der Zeichenfolge. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach:** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab der Zeichenfolge bis zum Ende der Daten. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Großschreibung ändern**: Ändert alle Zeichen in Ihren Zeichenfolgen zu Groß- oder Kleinschreibung.

Beispiel
Das Element **„LIST_ITEM“** ist als Datenblock und als Zuweisungsbereich definiert.

```
{
  "DELIVERYNOTE": {
    "LIST_CUSTOMER_INFO": {
      "CUSTOMER_INFO": {
        "CUSTOMER_NAME": "Customer A",
        "CUSTOMER_STREET_ADDRESS": "Test St",
        "CUSTOMER_POST_ADDRESS": "1234, Test City",
        "CUSTOMER_NUMBER": "1234",
        "CURRENCY": "EUR",
        "DELIVERY_METHOD": "Express delivery",
        "EDI_INFORMATION": "EDI",
        "ORDER_TYPE": "CSO",
        "ORDER_NUMBER": "123",
      }
    }
  }
}
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

2.5. Etiketten- und Druckernamen anhand von Eingabedaten einstellen

Normalerweise extrahieren Filter Werte aus empfangenen Daten und senden diese Werte zwecks Druck an Etikettenvariablen. In solchen Fällen sind Etiketten- oder Druckernamen fest in den Aktionen integriert. Beispielsweise wird die Aktion Etikett öffnen den Etikettennamen und die Aktion Drucker einstellen den Druckernamen fest codieren. Die Eingabedaten können jedoch auch *Metadaten* bereitstellen. Dies sind die Werte, die im Rahmen der Verarbeitung in NiceLabel Automation zwar genutzt, aber nicht auf das Etikett gedruckt werden, darunter Etikettenname, Druckername, Anzahl von Etiketten usw.

Um die Werte von Metadatenfeldern im Druckprozess zu nutzen, tun Sie Folgendes.

1. **Rekonfiguration des Filters:** Sie müssen neue Felder für die Eingabedaten definieren, damit auch die Metadatenfelder extrahiert werden.
2. **Definition der Variablen:** Sie müssen die Variablen, welche die Metadaten speichern, manuell definieren; sie sind nicht auf dem Etikett vorhanden und können nicht importiert werden. Verwenden Sie anschauliche Namen wie **`EtikettName`**, **`DruckerName`** und **`Menge`**. Sie können jeden beliebigen Namen für die Variablen verwenden.
3. **Rekonfiguration der Zuordnung:** Sie müssen die Aktion Datenfilter verwenden manuell konfigurieren, um die Metadatenfelder neuen Variablen zuzuordnen.
4. **Rekonfiguration der Aktion:** Konfigurieren Sie die Aktion **Etikett öffnen** neu, damit sie das durch die Variable **`EtikettName`** vorgegebene Etikett öffnet, und die Aktion ‚Drucker einstellen", damit der durch die Variable **`DruckerName`** vorgegebene Drucker verwendet wird.

Beispiel
Die CSV-Datei enthält Etikettendaten, aber auch *Metadaten* wie den Etiketten- und Druckernamen sowie die Anzahl von Etiketten. Der Filter für strukturierten Text extrahiert alle Felder, sendet Etikett-bezogene Werte an die Etikettenvariablen und nutzt die *Metadaten*, um die Aktionen „Etikett öffnen", „Drucker einstellen" und „Etikett drucken" zu konfigurieren.

```
label_name;label_count;printer_name;art_code;art_name;ean13;weight
label1.nlbl;1;CAB A3 203DPI;00265012;SAC.PESTO 250G;383860026501;1,1 kg
label2.nlbl;1;Zebra R-402;00126502;TAGLIOLINI 250G;383860026002;3,0 kg
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

3. Informationen zu Triggern

PRODUKTEBENEN-INFO

Diese Funktion ist nicht in jeder NiceLabel Automation Produktebene vollständig verfügbar.

NiceLabel Automation ist eine ereignisbasierte Anwendung, die die Ausführung von Aktionen zum Zeitpunkt von Änderungen in überwachten Ereignissen auslöst. Sie können die verfügbaren Trigger verwenden, um Änderungen in Ereignissen zu verfolgen, beispielsweise die Ablage von Dateien in einem bestimmten Ordner, den Datenempfang an einem bestimmten TCP/IP-Socket, HTTP-Nachrichten usw. Der Hauptzweck von Triggern besteht darin, Änderungen in Ereignissen zu erkennen, die vom Ereignis bereitgestellten Daten abzurufen und Aktionen auszuführen.

Die meisten Trigger warten passiv darauf, dass das überwachte Ereignis eintritt. Es gibt zwei Ausnahmen. **Datenbank-Trigger** ist ein aktiver Trigger, der die überwachte Datenbank regelmäßig auf Änderungen überprüft. **Trigger für die serielle Schnittstelle** kann entweder auf eine eingehende Verbindung warten oder aber die Schnittstelle in festgelegten Intervallen aktiv auf Daten abfragen.

Trigger verarbeiten

In den meisten Fällen erhält der Trigger Daten, die auf Etiketten gedruckt werden sollen. Nachdem der Trigger die Daten empfangen hat, werden die Aktionen in der vorgegebenen Reihenfolge von oben nach unten ausgeführt. Die empfangenen Daten können Werte für Etikettenobjekte enthalten. Bevor diese Werte genutzt werden können, müssen sie jedoch aus den empfangenen Daten extrahiert und in Variablen gespeichert werden. Die Filter definieren Extraktionsregeln. Bei Ausführung speichern die Filter die extrahierten Daten in den ihnen zugeordneten Variablen. Danach können Sie Aktionen ausführen, bei denen die Variablen genutzt werden, zum Beispiel „Etikett drucken“.

Wenn ein Ereignis eintritt, werden die bereitgestellten Eingabedaten in einer temporären Datei im **`%temp%`**-Ordner des Benutzers gespeichert. Die interne Variable **`DataFileName`** bezieht sich auf den Speicherort der temporären Datei. Nachdem der Trigger die Ausführung abgeschlossen hat, wird die Datei gelöscht.

Trigger-Eigenschaften

Um einen Trigger zu konfigurieren, müssen Sie die Methode des Datenempfangs und die auszuführenden Aktionen definieren. Optional können Sie auch Variablen verwenden. Die Trigger-Konfiguration besteht aus drei Abschnitten.

- **Einstellungen:** Definiert die Hauptparameter des ausgewählten Triggers. Wählen Sie das Ereignis aus, das der Trigger auf Veränderungen überwachen soll, oder legen Sie den Kanal für eingehende Kommunikation fest. Auf der Einstellungen-Registerkarte können Sie die Script-Programmierungs-Enging und die Sicherheitsoptionen auswählen. Die verfügbaren Optionen hängen von der Art des Triggers ab. Weitere Informationen finden Sie im Abschnitt Trigger-Typen.
- **Variablen:** Auf dieser Registerkarte werden die im Trigger benötigten Variablen definiert. Normalerweise importieren Sie Variablen aus den Etikettenvorlagen, sodass Sie sie den Feldern zuordnen können, die aus den eingehenden Daten extrahiert werden. Sie können auch Variablen für die interne Nutzung im Rahmen verschiedener Aktionen definieren; diese Variablen werden nicht an das Etikett gesendet. Weitere Informationen finden Sie in den Abschnitten Variablen.
- **Aktionen:** Auf dieser Registerkarte werden die Aktionen definiert, die ausgeführt werden sollen, sobald der Trigger Änderungen im überwachten Ereignis erkennt. Die Aktionen werden in der vorgegebenen Reihenfolge von oben nach unten ausgeführt. Weitere Informationen finden Sie im Abschnitt Aktionen..

Trigger-Typen

- **Dateitrigger:** Überwacht Änderungen in einer Datei oder einer Reihe von Dateien. Die Inhalte solcher Dateien können durch Filter geparst und in Aktionen verwendet werden.
- **Trigger für serielle Schnittstelle:** Überwacht die eingehende Kommunikation an der seriellen Schnittstelle RS232. Die Inhalte des Eingangsstroms werden durch Filter geparst und in Aktionen verwendet. Außerdem können die Daten auch in vorgegebenen Zeitintervallen vom externen Gerät abgerufen werden.
- **Datenbank-Trigger:** Überwacht Datensatz-Änderungen in SQL-Datenbanktabellen. Die Inhalte des ausgegebenen Daten-Sets können geparst und in Aktionen verwendet werden. Die Datenbank wird in festgelegten Intervallen geprüft. Zudem kann der Trigger die Datenbank nach Ausführung von Aktionen anhand von **`INSERT`**, **`UPDATE`** und **`INSERT SQL`**Anweisungen auch aktualisieren.
- **Planer-Trigger:** Führt Ihren Trigger in festen Zeitabständen aus.
- **TCP/IP Server Trigger:** Überwacht den eingehenden Rohdatenstrom, der am definierten Socket ankommt. Die Inhalte des Eingangsstroms werden durch Filter geparst und in Aktionen verwendet. Der TCP/IP Server Trigger kann bidirektional arbeiten und verwendet werden, um Feedback bereitzustellen.
- **TCP/IP Client Trigger:** Wandelt Ihr Automation in einen Daten empfangenden Client um, der sich mit TCP/IP-Servern verbindet.
- **HTTP Server Trigger:** Überwacht den eingehenden Datenstrom im HTTP-Format, der am definierten Socket ankommt. Die Inhalte des Eingangsstroms werden durch Filter geparst und in Aktionen verwendet. Benutzerauthentifizierung kann aktiviert werden. Ist bidirektional, stellt Feedback bereit.
- **Webdienst-Trigger:** Überwacht den eingehenden Datenstrom, der bei der definierten Webdienstmethode ankommt. Die Inhalte des Eingangsstroms werden durch Filter geparst und in Aktionen verwendet. Ist bidirektional, stellt Feedback bereit.
- **Cloud-Trigger:** Erfasst Daten aus NiceLabel Cloud.

Fehlerhandhabung in Triggern

- **Konfigurationsfehler:** Der Trigger weist einen Fehlerstatus auf, wenn er falsch oder unvollständig konfiguriert wurde. Dies ist beispielsweise der Fall, wenn Sie den Dateitrigger zwar konfiguriert, aber nicht den Namen der Datei angegeben haben, die auf Änderungen überwacht werden soll. Oder Sie haben die Aktion zum Drucken von Etiketten definiert, aber den Etikettennamen nicht angegeben. Sie können Trigger mit Konfigurationsfehlern zwar speichern, aber nicht in Automation Manager ausführen, bevor das Problem behoben wurde. Der in einer tieferen Ebene der Konfiguration gemeldete Fehler breitet sich bis in die höheren Ebenen aus, weswegen die Fehlerposition leicht zu finden ist.

Beispiel

Wenn Sie eine Aktion mit Fehlerstatus haben, zeigen alle Aktionen auf höherer Ebene den Fehlerstatus ebenfalls an. Das Fehlersymbol wird auf der Aktionen-Registerkarte und im Triggernamen angezeigt.

- **Sich überlappende Konfigurationen:** Obwohl eine Konfiguration durchaus mehrere Trigger enthalten kann, die dasselbe Ereignis überwachen (etwa dieselbe Datei) oder denselben TCP/IP-Port abhören, können solche Trigger nicht gleichzeitig ausgeführt werden. Wenn Sie den Trigger in Automation Manager starten, wird er nur ausgeführt, wenn kein anderer Trigger aus derselben oder einer anderen Konfiguration dasselbe Ereignis überwacht.

Feedback zum Status von Druckaufträgen
Weitere Informationen finden Sie im Abschnitt Feedback zum Status von Druckaufträgen.

3.1. Dateitrigger

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

Das Dateitrigger-Ereignis tritt ein, wenn:

- Änderungen an einer überwachten Datei eintreten
- eine Reihe von Daten im überwachten Verzeichnis geändert werden
- eine neue Datei zum überwachten Ordner hinzugefügt wird

Je nach der Trigger-Konfiguration informiert das Windows-Betriebssystem den Trigger über die geänderten Dateien, oder der Trigger selbst verwaltet eine Liste mit Zeitstempeln zu Dateiänderungen und löst aus, nachdem die Datei einen neueren Zeitstempel erhalten hat.

ANMERKUNG

Typische Nutzung: Das Geschäftssystem führt eine Transaktion durch, die wiederum eine Trigger-Datei in einem gemeinsam genutzten Ordner erzeugt. Der Dateninhalt kann strukturiert sein (CSV, XML und andere Formate) oder ein veraltetes Format aufweisen. In jedem Fall liest NiceLabel Automation die Daten aus, parst Werte anhand von Filtern und druckt diese Werte auf Etiketten. Weitere Informationen zum Parsen und Extrahieren von Daten finden Sie im Abschnitt Informationen zu Filtern.

TIPP

Hilfe bei der Erstellung von Konfigurationen mit dem Dateitrigger finden Sie in den Automation Beispieldateien: Zusammengesetzte CSV-Dateien, CSV Medium, CSV Simple, usw. Sie finden Beispieldateien unter **Hilfe** > **Beispieldateien**.

Allgemein

In diesem Bereich können Sie die wichtigsten Dateitrigger-Einstellungen vornehmen.

- **Name:** Gibt den eindeutigen Namen des Triggers an. Die Namen helfen Ihnen dabei, zwischen verschiedenen Triggern zu unterscheiden, wenn Sie sie in Automation Builder konfigurieren und später in Automation Manager ausführen.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Triggers zu beschreiben. Geben Sie für die Benutzer eine kurze Erklärung zur Funktionsweise des Triggers ein.
- **Angegebene Datei finden:** Gibt den Pfad und den Namen der Datei ein, die Sie auf Änderungen überwachen.
- **Einen Satz von Dateien im angegebenen Ordner finden:** Gibt den Pfad zu dem Ordner an, den Sie auf Dateiänderungen überwachen, sowie die Dateinamen. Sie können Standard-Windows-Platzhalter wie „*“ und „?“ verwenden. Einige Dateitypen sind in der Dropdown-Liste vordefiniert, aber Sie können auch eigene Typen angeben.

ANMERKUNG

Wenn Sie einen Netzwerkordner überwachen, sollten Sie auf jeden Fall die UNC-Notation `\\server\share\file` verwenden. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk.

- **Änderungen automatisch erkennen:** NiceLabel Automation antwortet auf Dateiänderungen, sobald die Datei erstellt oder geändert wird. In diesem Fall informiert das Windows-Betriebssystem den NiceLabel Automation Dienst über die Änderung. Sie können diese Option nutzen, wenn sich der überwachte Ordner auf der lokalen Festplatte befindet, und auch in einigen Netzwerkumgebungen.
- **In Intervallen auf Änderungen im Ordner prüfen:** NiceLabel Automation überprüft den Ordner in den festgelegten Zeitintervallen auf Dateiänderungen. In diesem Fall überwacht NiceLabel Automation

Ordner eigenständig auf Dateiänderungen. Diese Abfragemethode ist für gewöhnlich langsamer als die automatische Erkennung. Verwenden Sie sie nur, wenn die automatische Erkennung in Ihrer Umgebung nicht eingesetzt werden kann.

Ausführen

Die Optionen im Bereich **Dateizugriff** geben an, wie die Anwendung auf die Triggerdatei zugreift.

- **Datei exklusiv öffnen:** Öffnet die Trigger-Datei im exklusiven Modus. Auf diese Weise kann keine andere Anwendung gleichzeitig auf die Datei zugreifen. Dies ist die Standardauswahl.
- **Datei nur mit Leserecht öffnen:** Öffnet die Triggerdatei im Nur-Lesen-Modus.
- **Datei mit Lese- und Schreibrechten öffnen:** Öffnet die Triggerdatei im Lesen-Schreiben-Modus.
- **Intervall, in dem versucht wird, die Datei erneut zu öffnen:** Gibt das Zeitintervall an, nach dem NiceLabel Automation erneut versucht, die Triggerdatei zu öffnen. Falls der Zugriff auf die Datei nach dieser Zeit immer noch nicht möglich ist, gibt NiceLabel Automation einen Fehler aus.

Die Optionen im Bereich **Überwachungsoptionen** geben die Möglichkeiten zur Dateierkennung an.

- **Dateigröße überprüfen:** Aktiviert die Erkennung von Änderungen nicht nur anhand des Datei-Zeitstempels, sondern auch anhand der Dateilänge. Die Änderungen im Datei-Zeitstempel könnten unter bestimmten Umständen nicht erkannt werden. Daher prüft Automation auf Änderungen der Dateigröße und löst die Aktionen aus.
- **Leere Triggerdateien ignorieren:** Wenn die Triggerdatei keinen Inhalt hat, wird sie ignoriert. Die Aktionen werden nicht ausgeführt.
- **Triggerdatei löschen:** Nach Erkennung der Änderung in der Triggerdatei und Auslösen des Triggers löscht Automation die Datei. Durch Aktivierung dieser Option können verarbeitete Dateien aus dem Ordner entfernt werden.

ANMERKUNG

NiceLabel Automation Behält immer eine Sicherungskopie der empfangenen Triggerdaten. Der Inhalt der Triggerdatei wird unter einem eindeutigen Dateinamen gespeichert. Dies ist wichtig, wenn Sie den Inhalt der Triggerdatei in einigen Aktionen wie **Führe Befehlsdatei aus** benötigen. Auf den Speicherort der Backup-Triggerdaten wird von der internen Variablen *DataFileName* verwiesen.

- **Datei leeren:** Nach Ausführung der Aktionen wird die Triggerdatei geleert. Dies ist nützlich, wenn dritte Anwendungen Daten in die Triggerdatei schreiben. In solchen Fällen möchten Sie die Datei behalten, damit in sie geschrieben werden kann, wollen aber die alten Daten nicht drucken.
- **Änderungen nachverfolgen wenn Trigger inaktiv ist:** Löst den Trigger für Dateien aus, an denen Änderungen vorgenommen werden, während der Trigger inaktiv ist. Falls Ihr NiceLabel Automation nicht in eine Hochverfügbarkeitsumgebung mit Backup-Servern eingebunden ist, könnten die eingehenden Triggerdateien bei einem Serverausfall verloren gehen. Wenn NiceLabel Automation wieder online ist, werden die vorhandenen Triggerdateien verarbeitet.

- **Anzahl der simultan stattfindenden Maßnahmen**: Führen Sie die Anzahl Ihrer simultan stattfindenden Maßnahmen auf. Sie können mit der nächsten auch schon beginnen, bevor die vorhergehende Maßnahme abgeschlossen wurde. Die Verarbeitung Ihrer Maßnahmen-Reihenfolge bleibt dabei gleich, während dieselbe Maßnahme von einem anderen Thread simultan mit der Ausführung beginnen kann.
  Die Höchstzahl Ihrer simultan stattfindenden Maßnahmen hängt auch von Ihrer Hardwareleistung ab. Mehr darüber erfahren Abschnitt 7.1, „Parallele Verarbeitung“.

Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikationsparameter fest, durch die Sie Feedback von der Druck-Engine erhalten können.

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

- **Überwachtes Drucken:** Aktiviert den synchronen Druckmodus. Verwenden Sie ihn, wenn Sie den Status des Druckauftrags an die Drittanwendung zurücksenden wollen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Die Optionen im Abschnitt **Datenverarbeitung** geben an, ob Sie die Daten kürzen möchten, damit sie in eine Variable passen, oder ob die fehlenden Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet Fehler und unterbricht den Druckprozess, wenn Sie versuchen, zu lange Werte in Etikettenvariablen zu speichern oder Werte für nicht vorhandene Etikettenvariablen anzugeben.

- **Übermäßig lange Variableninhalte ignorieren:** kürzt Datenwerte, die die Länge der Variable gemäß Definition im Etiketten-Designer überschreiten, um sie auf passende Länge zu bringen. Diese Option wird wirksam, wenn Sie Variablenwerte in Filtern oder aus Befehlsdateien festlegen und wenn Sie Werte von Triggervariablen den gleichnamigen Etikettenvariablen zuordnen.

Beispiel

Die Etikettenvariable akzeptiert maximal 5 Zeichen. Wenn diese Option aktiviert ist, wird jeder Wert über 5 Zeichen auf die ersten 5 Zeichen gekürzt. Wenn der Wert 1234567 lautet, ignoriert die 6. Und 7. Stelle.

- **Fehlende Etikettenvariablen ignorieren:** Beim Drucken mit Befehlsdateien (z. B. JOB-Dateien), werden alle Variablen ignoriert, die:
  - in der Befehlsdatei angegeben sind (anhand des SET Befehls)
  - nicht auf dem Etikett definiert sind

  Etwas Ähnliches geschieht, wenn Sie einen Zuweisungsbereich in einem Filter definieren, um alle Name-Wert-Paare zu extrahieren, Ihr Etikett jedoch weniger Variablen enthält.
  Wenn Sie Werte von nicht vorhandenen Etikettenvariablen einstellen, gibt einen Fehler aus. Wenn diese Option aktiviert ist, wird der Druck fortgesetzt.

Die Optionen im Abschnitt **Scripting** geben die Scripting-Möglichkeiten an.

- **Scripting-Sprache:** Wählt die Scripting-Sprache für den Trigger aus. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden die ausgewählte Scripting-Sprache.

Die Optionen im Abschnitt **Empfangene Daten speichern** legen die Befehle fest, die für vom Trigger empfangene Daten zur Verfügung stehen.

- **Vom Trigger empfangene Daten speichern in Datei:** Aktivieren Sie diese Option, um vom Trigger empfangene Daten zu speichern. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und den Dateinamen enthält.
- **Bei Fehler vom Trigger empfangene Daten speichern in Datei:** Aktivieren sie diese Option, um die vom Trigger empfangenen Daten nur dann zu speichern, wenn ein Fehler bei der Ausführung der Aktion auftritt. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu behalten, die das Problem hervorgerufen haben, um die Lösungsfindung zu erleichtern.

ANMERKUNG

Stellen Sie sicher, dass Sie die Unterstützung für Überwachtes Drucken aktivieren. Ist sie nicht aktiviert, kann Fehler bei der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

ANMERKUNG

speichert die empfangenen Daten in einer temporären Datei. Diese temporäre Datei wird sofort nach Abschluss der Trigger-Ausführung gelöscht. Die interne Variable **`DataFileName`** verweist auf diesen Dateinamen. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

Sicherheit

- **Trigger sperren und verschlüsseln:** Aktiviert den Trigger-Schutz. Wenn Sie die Option aktivieren, wird der Trigger gesperrt und kann nicht mehr bearbeitet werden. Dadurch werden die Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

3.2. Trigger für serielle Schnittstelle

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

Das Triggerereignis für die serielle Schnittstelle tritt ein, wenn Daten an der überwachten seriellen Schnittstelle RS232 empfangen werden.

Typische Nutzung: **(1) Druckerersatz.** Sie ersetzen den bisherigen, über die serielle Schnittstelle verbundenen Etikettendrucker. An dessen Stelle nimmt NiceLabel Automation die Daten an, extrahiert die Werte für Etiketten aus dem empfangenen Druckstrom und erstellt einen Druckauftrag für das neue Druckermodell. **(2) Waagen.** Waagen geben Daten zu einem gewogenen Objekt aus. NiceLabel Automation

extrahiert die erforderlichen Daten aus dem empfangenen Datenstrom und druckt ein Etikett. Weitere Informationen zum Parsen und Extrahieren von Daten finden Sie im Abschnitt Informationen zu Filtern.

TIPP

Hilfe beim Erstellen von Konfigurationen mit dem Serial Port Trigger finden Sie in der Automation Beispieldatei “Scan & Print from Excel“. Sie finden Beispieldateien unter **Hilfe** > **Beispieldateien**.

Allgemein

In diesem Bereich können Sie die wichtigsten Dateitrigger-Einstellungen vornehmen.

- **Name:** Gibt den eindeutigen Namen des Triggers an. Die Namen helfen Ihnen dabei, zwischen verschiedenen Triggern zu unterscheiden, wenn Sie sie in Automation Builder konfigurieren und später in Automation Manager ausführen.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Triggers zu beschreiben. Geben Sie für die Benutzer eine kurze Erklärung zur Funktionsweise des Triggers ein.
- **Anschluss:** Gibt die Nummer der seriellen Schnittstelle (COM) an, an der die eingehenden Daten empfangen werden. Verwenden Sie eine Schnittstelle, die nicht von einer anderen Anwendung oder einem anderen Gerät genutzt wird, etwa von einem Druckertreiber. Falls die ausgewählte Schnittstelle in Verwendung ist, können Sie den Trigger nicht in Automation Manager starten.

Die Optionen im Bereich **Schnittstellen-Einstellungen** geben die Kommunikationsparameter an, die den Parametern des Geräts an der seriellen Schnittstelle entsprechen müssen.

- **Port-Initialisierung deaktivieren:** Gibt an, dass beim Start des Triggers in Automation Manager keine Schnittstellen-Initialisierung durchgeführt wird. Diese Option ist manchmal für virtuelle COM-Schnittstellen erforderlich.

Ausführen

- **Initialisierungsdaten verwenden:** Gibt an, dass die Initialisierungs-Zeichenfolge bei jedem Start des Triggers an das Gerät an der seriellen Schnittstelle gesendet werden soll. Einige serielle Geräte müssen aufgeweckt oder in den Standby-Modus versetzt werden, bevor sie die Daten bereitstellen können. Weitere Informationen über die Initialisierungs-Zeichenfolge sowie dazu, unter welchen Umständen Sie sie benötigen, finden Sie im Benutzerhandbuch Ihres Geräts. Sie können binäre Zeichen einschließen. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen.
- **Datenpolling verwenden:** Gibt an, dass der Trigger das Gerät aktiv auf Daten abfragen soll. Der Trigger sendet die im Feld **Inhalt** angegebenen Befehle zu den vorgegebenen Zeitintervallen. Dieses Feld kann binäre Zeichen enthalten. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen.

Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikationsparameter fest, durch die Sie Feedback von der Druck-Engine erhalten können.

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

- **Überwachtes Drucken:** Aktiviert den synchronen Druckmodus. Verwenden Sie ihn, wenn Sie den Status des Druckauftrags an die Drittanwendung zurücksenden wollen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Die Optionen im Abschnitt **Datenverarbeitung** geben an, ob Sie die Daten kürzen möchten, damit sie in eine Variable passen, oder ob die fehlenden Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet Fehler und unterbricht den Druckprozess, wenn Sie versuchen, zu lange Werte in Etikettenvariablen zu speichern oder Werte für nicht vorhandene Etikettenvariablen anzugeben.

- **Übermäßig lange Variableninhalte ignorieren:** kürzt Datenwerte, die die Länge der Variable gemäß Definition im Etiketten-Designer überschreiten, um sie auf passende Länge zu bringen. Diese Option wird wirksam, wenn Sie Variablenwerte in Filtern oder aus Befehlsdateien festlegen und wenn Sie Werte von Triggervariablen den gleichnamigen Etikettenvariablen zuordnen.

  Beispiel

  Die Etikettenvariable akzeptiert maximal 5 Zeichen. Wenn diese Option aktiviert ist, wird jeder Wert über 5 Zeichen auf die ersten 5 Zeichen gekürzt. Wenn der Wert 1234567 lautet, ignoriert die 6. Und 7. Stelle.

- **Fehlende Etikettenvariablen ignorieren:** Beim Drucken mit Befehlsdateien (z. B. JOB-Dateien), werden alle Variablen ignoriert, die:
  - in der Befehlsdatei angegeben sind (anhand des SET Befehls)
  - nicht auf dem Etikett definiert sind

  Etwas Ähnliches geschieht, wenn Sie einen Zuweisungsbereich in einem Filter definieren, um alle Name-Wert-Paare zu extrahieren, Ihr Etikett jedoch weniger Variablen enthält.
  Wenn Sie Werte von nicht vorhandenen Etikettenvariablen einstellen, gibt einen Fehler aus. Wenn diese Option aktiviert ist, wird der Druck fortgesetzt.

Die Optionen im Abschnitt **Scripting** geben die Scripting-Möglichkeiten an.

- **Scripting-Sprache:** Wählt die Scripting-Sprache für den Trigger aus. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden die ausgewählte Scripting-Sprache.

Die Optionen im Abschnitt **Empfangene Daten speichern** legen die Befehle fest, die für vom Trigger empfangene Daten zur Verfügung stehen.

- **Vom Trigger empfangene Daten speichern in Datei:** Aktivieren Sie diese Option, um vom Trigger empfangene Daten zu speichern. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und den Dateinamen enthält.

- **Bei Fehler vom Trigger empfangene Daten speichern in Datei:** Aktivieren sie diese Option, um die vom Trigger empfangenen Daten nur dann zu speichern, wenn ein Fehler bei der Ausführung der Aktion auftritt. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu behalten, die das Problem hervorgerufen haben, um die Lösungsfindung zu erleichtern.

  ANMERKUNG

  Stellen Sie sicher, dass Sie die Unterstützung für Überwachtes Drucken aktivieren. Ist sie nicht aktiviert, kann Fehler bei der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

  ANMERKUNG

  speichert die empfangenen Daten in einer temporären Datei. Diese temporäre Datei wird sofort nach Abschluss der Trigger-Ausführung gelöscht. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

Sicherheit

- **Trigger sperren und verschlüsseln:** Aktiviert den Trigger-Schutz. Wenn Sie die Option aktivieren, wird der Trigger gesperrt und kann nicht mehr bearbeitet werden. Dadurch werden die Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

3.3. Datenbank-Trigger

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

Ein Datenbank-Triggerereignis tritt ein, wenn eine Änderung an der überwachten Datenbanktabelle erkannt wird. Das kann der Fall sein, wenn neue Datensätze vorhanden sind oder vorhandene Datensätze aktualisiert wurden. Der Datenbank-Trigger wartet nicht auf Ereignisänderungen wie z. B. Datenlieferungen. Stattdessen ruft er Daten aus der Datenbank in vorgegebenen Zeitintervallen ab.

Typische Nutzung: Ein vorhandenes Geschäftssystem führt eine Transaktion durch, wodurch wiederum Daten in einer Datenbanktabelle aktualisiert werden. NiceLabel Automation erkennt die aktualisierten und neuen Datensätze und druckt ihren Inhalt auf Etiketten.

Allgemein

In diesem Bereich können Sie die wichtigsten Dateitrigger-Einstellungen vornehmen.

- **Name:** Gibt den eindeutigen Namen des Triggers an. Die Namen helfen Ihnen dabei, zwischen verschiedenen Triggern zu unterscheiden, wenn Sie sie in Automation Builder konfigurieren und später in Automation Manager ausführen.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Triggers zu beschreiben. Geben Sie für die Benutzer eine kurze Erklärung zur Funktionsweise des Triggers ein.

- **Datenbankverbindung:** Gibt die Zeichenfolge für die Verbindung zur Datenbank an. Klicken Sie auf **Definieren**, um das Datenbank-Dialogfeld zu öffnen. Nutzen Sie es, um die Datenbankverbindung zu konfigurieren, einschließlich Datenbanktyp, Tabellenname und Benutzer-Zugangsdaten. Sie müssen eine Verbindung zu einer Datenbank herstellen, die Zugriff über SQL-Befehle unterstützt. Daher können Sie den Datenbank-Trigger nicht für die automatische Erkennung von Daten in CSV-Textdateien und Microsoft Excel Tabellenkalkulationen verwenden.

  > **ANMERKUNG**
  > Die Konfigurationsdetails hängen von der Art der ausgewählten Datenbank ab. Die Optionen im Dialogfeld hängen vom genutzten Datenbanktreiber ab. Konfigurationsdetails finden Sie im Benutzerhandbuch für Ihren Datenbanktreiber. Weitere Informationen zu Datenbankverbindungen finden Sie in Abschnitt Zugriff auf Datenbanken.

- **Datenbank in folgenden Zeitintervallen prüfen:** Gibt die Zeitintervalle an, in denen die Datenbank auf Änderungen in Datensätzen abgefragt wird.
- **Erkennungsoptionen und Erweitert:** Diese Optionen ermöglichen Ihnen eine Feinabstimmung des Erkennungsmechanismus für Datensätze. Nach Abruf der Datensätze aus der Datenbank wird automatisch die Aktionen-Registerkarte für die Aktion „Für jeden Datensatz“ angezeigt, mit der Sie Tabellenfelder zu Etikettenvariablen zuordnen können.

  Alle auf dem einzigartigen Feldwert basierenden Datensätze übernehmen
  Wenn diese Option aktiviert ist, überwacht der Trigger das angegebene autoinkrementelle numerische Feld in der Tabelle. NiceLabel Automation speichert den Feldwert für den letzten verarbeiteten Datensatz. Beim nächsten Abfrageintervall werden nur Datensätze abgerufen, deren Werte höher sind als der gespeicherte Wert.

  Um diese Option zu konfigurieren, müssen Sie den Namen der Tabelle, in der sich die Datensätze befinden **`(Tabellenname)`**, das autoinkrementelle Feld **`(Schlüsselfeld)`** sowie den Startwert für das Feld **`(Standardwert des Schlüsselfeldes)`** auswählen. Die Variable **`KeyField`** wird intern verwendet, um auf den letzten bekannten Wert des Schlüsselfelds zu verweisen.

  > **ANMERKUNG**
  > Der letzte Wert des Schlüsselfelds wird intern gespeichert, aber in der Konfiguration nicht aktualisiert; daher ändert sich der Wert für **`Standardwert des Schlüsselfeldes`** in diesem Dialogfeld nicht. Sie können die Konfiguration sicher neu laden und/oder diesen Trigger im Automation Manager anhalten/starten, ohne dass der letzte bekannte Wert verloren geht. Wenn Sie jedoch die Konfiguration aus Automation Manager entfernen und wieder hinzufügen, wird der letzte bekannte Wert des Schlüsselfelds auf den Wert zurückgesetzt, den Sie unter **`Standardwert des Schlüsselfeldes`** angegeben haben.

Datensätze übernehmen und löschen

Wenn diese Option aktiviert ist, werden alle Datensätze aus der Tabelle abgerufen und gelöscht. Um diese Option zu konfigurieren, wählen Sie den Namen der Tabelle aus, in der sich die Datensätze befinden (**`Tabellenname`**), und geben den primären Schlüssel in der Tabelle (**`Schlüsselfelder`**) an. Obwohl eine Tabelle in Automation nicht zwangsläufig über einen primären Schlüssel verfügen muss, ist es sehr empfehlenswert, einen solchen zu definieren. Ist ein primärer Schlüssel vorhanden, werden die Datensätze nacheinander gelöscht, sobald der jeweilige Datensatz in den Aktionen verarbeitet wird.

WARNUNG

Ist kein primärer Schlüssel vorhanden, werden alle vom aktuellen Trigger empfangenen Datensätze auf einmal gelöscht. Dies ist in Ordnung, solange bei der Datensatzverarbeitung kein Fehler auftritt. Wenn jedoch bei einem Datensatz ein Verarbeitungsfehler auftritt, unterbricht Automation die Verarbeitung weiterer Datensätze. Da alle in diesem Abrufintervall erfassten Datensätze bereits gelöscht wurden, ohne zuvor verarbeitet worden zu sein, könnten Sie Daten verlieren. Daher empfiehlt sich die Definition eines primären Schlüssels in einer Tabelle.

Beispiele für SQL-Code

ANMERKUNG

Die folgenden SQL-Anweisungen können nur gelesen werden und werden ausschließlich zu Referenzzwecken angegeben. Um benutzerdefinierte SQL-Anweisungen bereitzustellen, wählen Sie die Erkennungsmethode **Datensätze mit SQL Anweisung übernehmen und bearbeiten**.

Beispiel für eine Tabelle:

ID	ProductID	CodeEAN	ProductDesc	AlreadyPrinted
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

Beispiel einer SQL-Aktualisierungsanweisung, falls die Tabelle einen primären Index enthält:

```
DELETE FROM [Table]
WHERE [ID] = :ID
```

Das **`ID`**Feld in der Tabelle ist als primärer Index definiert. Das Konstrukt **`:ID`** im WHERE-Abschnitt enthält den Wert des ID-Felds in jeder Iteration. Für den ersten Datensatz ist der Wert von **`ID`** gleich 1, für den zweiten Datensatz gleich 2 usw. Durch Verwendung des Doppelpunkts vor dem Feldnamen in der SQL-Anweisung wird vorgegeben, wie die Variable genutzt wird.

Beispiel einer SQL-Aktualisierungsanweisung, falls für die Tabelle kein primärer Index definiert ist:

```
DELETE FROM [Table]
```

Wenn kein primärer Index für die Tabelle definiert ist, werden alle Datensätze daraus gelöscht, nachdem der erste Datensatz verarbeitet wurde.

Datensätze übernehmen und aktualisieren

Wenn diese Option aktiviert ist, werden alle Datensätze aus der Tabelle abgerufen und aktualisiert. Sie können einen individuellen Wert in ein Feld der Tabelle schreiben, um anzuzeigen, dass der jeweilige Datensatz bereits gedruckt wurde. Um diese Option zu konfigurieren, müssen Sie den Namen der Tabelle, in der sich die Datensätze befinden **`(Tabellenname)`** und das zu aktualisierende Feld **`(Aktualisierungsfeld)`** auswählen und den im Feld zu speichernden Wert **`(Aktualisierungswert)`** eingeben. Intern wird die Variable **`UpdateValue`** in der SQL-Anweisung verwendet, um sich auf den aktuellen Wert des Felds zu beziehen (**`Aktualisierungswert`**).

Obwohl Automation eine Tabelle in nicht zwangsläufig über einen primären Schlüssel verfügen muss, ist es sehr empfehlenswert, einen solchen zu definieren. Ist ein primärer Schlüssel vorhanden, werden die Datensätze nacheinander aktualisiert, sobald der jeweilige Datensatz in den Aktionen verarbeitet wird.

WARNUNG

Ist kein primärer Schlüssel vorhanden, werden alle im Trigger empfangenen Datensätze auf einmal aktualisiert. Dies ist unproblematisch, solange keine Fehler bei der Verarbeitung der Datensätze auftreten. Tritt aber bei der Verarbeitung eines Datensatzes ein Fehler auf, beendet die Automation die Verarbeitung weiterer Datensätze. Da alle in diesem Abrufintervall erfassten Datensätze bereits aktualisiert wurden, ohne zuvor verarbeitet worden zu sein, könnten Sie Daten verlieren. Daher empfiehlt sich die Definition eines primären Schlüssels in einer Tabelle.

Beispiele für SQL-Code

ANMERKUNG

Die folgenden SQL-Anweisungen können nur gelesen werden und werden ausschließlich zu Referenzzwecken angegeben. Um benutzerdefinierte SQL-Anweisungen bereitzustellen, wählen Sie die Erkennungsmethode **Datensätze mit SQL Anweisung übernehmen und bearbeiten**.

Beispiel für eine Tabelle:

ID	ProductID	CodeEAN	ProductDesc	AlreadyPrinted
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

Beispiel einer SQL-Aktualisierungsanweisung, wenn die Tabelle einen primären Index enthält:

```
UPDATE [Table]
SET [AlreadyPrinted] = :UpdateValue
WHERE [ID] = :ID
```

Das **ID**Feld in der Tabelle ist als primärer Index definiert. Das Konstrukt **:ID** im WHERE-Abschnitt enthält den Wert des ID-Felds in jeder Iteration. Für den ersten Datensatz ist der Wert von **ID** gleich 1, für den zweiten Datensatz gleich 2 usw. Durch Verwendung eines Doppelpunkts vor dem Feldnamen in der SQL-Anweisung wird die Nutzung einer Variablen vorgegeben. Das Feld **UpdateValue** wird in der Trigger-Konfiguration anhand des Bearbeitungsfelds **Aktualisierungswert** definiert.

Beispiel einer SQL-Aktualisierungsanweisung, wenn für die Tabelle kein primärer Index definiert ist:

```
UPDATE [Table]
SET [AlreadyPrinted] = :UpdateValue
```

Wenn kein primärer Index für die Tabelle definiert ist, werden alle Datensätze daraus aktualisiert, nachdem der erste Datensatz verarbeitet wurde.

Datensätze mit SQL Anweisung übernehmen und bearbeiten
In diesem Fall ist die Erstellung von SQL-Anweisungen zur Datensatzextraktion und Feldaktualisierung vollständig Ihnen überlassen. Um diese Option zu konfigurieren, müssen Sie eine individuelle SQL-Anweisung für den Abruf von Datensätzen **`(SQL-Suchanweisung)`** und eine individuelle SQL-Anweisung für die Aktualisierung der Datensätze nach der Verarbeitung **`(SQL-Aktualisierungsanweisung)`** angeben. Klicken Sie auf die Schaltfläche **Test**, um Ihre SQL-Anweisungen testweise auszuführen und das Ergebnis auf dem Bildschirm anzuzeigen.

Sie können Tabellenfeldwerte oder Werte von Triggervariablen als Parameter im WHERE-Abschnitt der SQL-Anweisung verwenden. Dazu würden Sie dem Feld- oder Variablennamen einen Doppelpunkt voranstellen (:). Dadurch verwendet NiceLabel Automation den aktuellen Wert dieses Felds bzw. dieser Variablen.

Beispiele für SQL-Code
Beispiel für eine Tabelle:

ID	ProductID	CodeEAN	ProductDesc	AlreadyPrinted
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

Beispiel für eine SQL-Suchanweisung:

Um die Datensätze abzurufen, die nicht bereits gedruckt wurden, führen Sie die folgenden Schritte aus. Das Feld **`AlreadyPrinted`** darf weder den Wert **`Y`** noch einen leeren oder NULL-Wert enthalten.

```
SELECT * FROM Table
WHERE AlreadyPrinted <> 'Y' or AlreadyPrinted is NULL
```

Aus der oben genannten Beispieltabelle werden zwei Datensätze mit den ID-Werten 2 und 3 extrahiert. Der erste Datensatz wurde bereits gedruckt und wird ignoriert.

Beispiel einer SQL-Aktualisierungsanweisung:

So markieren Sie bereits gedruckte Datensätze mit dem Wert **`Y`** im Feld **`AlreadyPrinted`**:

```
UPDATE [Table]
SET [AlreadyPrinted] = 'Y'
WHERE [ID] = :ID
```

Sie müssen einen Doppelpunkt (**`:`**) vor den Variablennamen in Ihrer SQL-Anweisung setzen, damit die Variable als solche erkannt werden kann. Sie können für Parameter im WHERE-Abschnitt jedes beliebige Feld aus der Tabelle verwenden. In diesem Beispiel aktualisieren wir das Feld **`AlreadyPrinted`** nur für den momentan verarbeiteten Datensatz (der Wert des Feldes **`ID`** muss mit dem Wert aus dem aktuellen Datensatz übereinstimmen). Auf ähnliche Weise können Sie auf andere Felder im Datensatz als **`:ProductID`** oder **`:CodeEAN`** verweisen, oder auch auf Variablen verweisen, die in diesem Datenbank-Trigger definiert sind.

So löschen Sie den aktuellen Datensatz aus der Tabelle:

```
DELETE FROM [Table]
WHERE [ID] = :ID
```

SQL Anweisung anzeigen: Erweitern Sie diesen Abschnitt, um die erzeugte SQL-Anweisung anzuzeigen und Ihre eigene Anweisung zu schreiben, wenn Sie die Option **Datensätze mit SQL Anweisung übernehmen und bearbeiten** ausgewählt haben.

Vorschau der SQL-Ausführung anzeigen

Um die Ausführung der SQL-Anweisungen zu testen und ihre Auswirkungen anzuzeigen, klicken Sie auf „Test“ in der Werkzeugleiste des SQL-Bearbeitungsbereichs. Der Abschnitt „Datenvorschau“ wird im rechten Bereich geöffnet. Klicken Sie auf die Schaltfläche **Ausführen**, um den SQL-Code auszuführen. Wenn Sie Werte aus einem Tabellenfeld in der SQL-Anweisung nutzen (mit einem Doppelpunkt (:) vor dem Feldnamen), müssen Sie Testwerte für sie angeben.

ANMERKUNG

Wenn Sie die Datenvorschau geöffnet und gerade einige Variablen zum Skript hinzugefügt haben, klicken Sie zweimal auf die **Test**-Schaltfläche. Dadurch wird der Datenvorschau-Bereich geschlossen und geöffnet und die Liste von Variablen in der Vorschau aktualisiert.

- **Ausführung simulieren:** Gibt an, dass alle Änderungen an der Datenbank ignoriert werden. Die Datenbank-Transaktion wird rückgängig gemacht, sodass keine Aktualisierungen in die Datenbank geschrieben werden.

Ausführen

Die Optionen unter „Ausführen“ geben an, wann die Datenbankaktualisierung stattfinden soll. Die Art der Aktualisierung hängt von den Erkennungsoptionen für den Trigger ab.

- **Vor Ausführung der Aktionen:** Gibt an, dass Datensätze vor Beginn der Ausführung der für diesen Trigger definierten Aktionen aktualisiert werden.
- **Nach Ausführung der Aktionen:** Gibt an, dass Datensätze nach Ausführung der für diesen Trigger definierten Aktionen aktualisiert werden. Normalerweise wollen Sie die Datensätze nach erfolgreicher Verarbeitung aktualisieren.

ANMERKUNG

Falls nötig, können Sie die Datensätze aktualisieren, während die Aktionen noch ausgeführt werden. Weitere Informationen finden Sie im Abschnitt SQL-Anweisung ausführen.

Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikationsparameter fest, durch die Sie Feedback von der Druck-Engine erhalten können.

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

- **Überwachtes Drucken:** Aktiviert den synchronen Druckmodus. Verwenden Sie ihn, wenn Sie den Status des Druckauftrags an die Drittanwendung zurücksenden wollen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Die Optionen im Abschnitt **Datenverarbeitung** geben an, ob Sie die Daten kürzen möchten, damit sie in eine Variable passen, oder ob die fehlenden Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet Fehler und unterbricht den Druckprozess, wenn Sie versuchen, zu lange Werte in Etikettenvariablen zu speichern oder Werte für nicht vorhandene Etikettenvariablen anzugeben.

- **Übermäßig lange Variableninhalte ignorieren:** kürzt Datenwerte, die die Länge der Variable gemäß Definition im Etiketten-Designer überschreiten, um sie auf passende Länge zu bringen. Diese Option wird wirksam, wenn Sie Variablenwerte in Filtern oder aus Befehlsdateien festlegen und wenn Sie Werte von Triggervariablen den gleichnamigen Etikettenvariablen zuordnen.

  Beispiel

  Die Etikettenvariable akzeptiert maximal 5 Zeichen. Wenn diese Option aktiviert ist, wird jeder Wert über 5 Zeichen auf die ersten 5 Zeichen gekürzt. Wenn der Wert 1234567 lautet, ignoriert die 6. Und 7. Stelle.

- **Fehlende Etikettenvariablen ignorieren:** Beim Drucken mit Befehlsdateien (z. B. JOB-Dateien), werden alle Variablen ignoriert, die:
  - in der Befehlsdatei angegeben sind (anhand des SET Befehls)
  - nicht auf dem Etikett definiert sind

  Etwas Ähnliches geschieht, wenn Sie einen Zuweisungsbereich in einem Filter definieren, um alle Name-Wert-Paare zu extrahieren, Ihr Etikett jedoch weniger Variablen enthält.
  Wenn Sie Werte von nicht vorhandenen Etikettenvariablen einstellen, gibt einen Fehler aus. Wenn diese Option aktiviert ist, wird der Druck fortgesetzt.

Die Optionen im Abschnitt **Scripting** geben die Scripting-Möglichkeiten an.

- **Scripting-Sprache:** Wählt die Scripting-Sprache für den Trigger aus. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden die ausgewählte Scripting-Sprache.

Die Optionen im Abschnitt **Empfangene Daten speichern** legen die Befehle fest, die für vom Trigger empfangene Daten zur Verfügung stehen.

- **Vom Trigger empfangene Daten speichern in Datei:** Aktivieren Sie diese Option, um vom Trigger empfangene Daten zu speichern. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und den Dateinamen enthält.
- **Bei Fehler vom Trigger empfangene Daten speichern in Datei:** Aktivieren sie diese Option, um die vom Trigger empfangenen Daten nur dann zu speichern, wenn ein Fehler bei der Ausführung der Aktion auftritt. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu behalten, die das Problem hervorgerufen haben, um die Lösungsfindung zu erleichtern.

> **ANMERKUNG**
> Stellen Sie sicher, dass Sie die Unterstützung für Überwachtes Drucken aktivieren. Ist sie nicht aktiviert, kann Fehler bei der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

> **ANMERKUNG**
> speichert die empfangenen Daten in einer temporären Datei. Diese temporäre Datei wird sofort nach Abschluss der Trigger-Ausführung gelöscht. Die interne Variable **`DataFileName`** verweist auf diesen Dateinamen. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

Sicherheit

- **Trigger sperren und verschlüsseln:** Aktiviert den Trigger-Schutz. Wenn Sie die Option aktivieren, wird der Trigger gesperrt und kann nicht mehr bearbeitet werden. Dadurch werden die Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

3.4. TCP/IP Server Trigger

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

Das TCP/IP-Server-Triggerereignis tritt ein, wenn das überwachte Socket (IP-Adresse und Portnummer) Daten empfängt.

Typische Nutzung: Typische Nutzung: Ein vorhandenes Geschäftssystem führt eine Transaktion durch, wodurch wiederum Daten über ein bestimmtes Socket an den NiceLabel Automation Server gesendet werden. Der Dateninhalt kann strukturiert sein (CSV, XML und andere Formate) oder ein veraltetes Format aufweisen. In jedem Fall liest NiceLabel Automation die Daten aus, parst Werte anhand von Filtern und druckt diese Werte auf Etiketten. Weitere Informationen zum Parsen und Extrahieren von Daten finden Sie im Abschnitt Informationen zu Filtern.

Allgemein

> **ANMERKUNG**
> Dieser Trigger unterstützt Internet Protocol Version 6 (IPv6).

In diesem Bereich können Sie die wichtigsten Dateitrigger-Einstellungen vornehmen.

- **Name:** Gibt den eindeutigen Namen des Triggers an. Die Namen helfen Ihnen dabei, zwischen verschiedenen Triggern zu unterscheiden, wenn Sie sie in Automation Builder konfigurieren und später in Automation Manager ausführen.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Triggers zu beschreiben. Geben Sie für die Benutzer eine kurze Erklärung zur Funktionsweise des Triggers ein.
- **Anschluss:** Gibt die Nummer der Schnittstelle an, wo eingehende Daten empfangen werden. Verwenden Sie die Portnummer, die noch nicht von einer anderen Anwendung genutzt wird. Falls die ausgewählte Schnittstelle in Verwendung ist, können Sie den Trigger nicht in Automation Manager starten. Weitere Informationen zu Sicherheitsüberlegungen finden Sie in Abschnitt Zugriff auf Ihre Trigger sichern.

ANMERKUNG

Wenn auf Ihrem Server Multi-Homing aktiviert ist (mehrere IP-Adressen auf einer oder mehreren Netzwerkkarten), antwortet NiceLabel Automation an der festgelegten Schnittstelle für alle IP-Adressen.

- **Maximale Anzahl gleichzeitig akzeptierter Verbindungen:** Gibt die maximale Anzahl gleichzeitig akzeptierter Verbindungen an. Dies definiert die Anzahl von Clients, die gleichzeitig Daten an den Trigger senden können.

Die Optionen im Abschnitt **Ausführungsereignis** geben vor, wann der Trigger auslösen und Aktionen ausführen soll.

- **Bei Trennung des Clients:** Gibt an, dass der Trigger auslöst, nachdem der Client Daten gesendet und die Verbindung geschlossen hat. Dies ist die Standardeinstellung.

ANMERKUNG

Falls Sie den Status des Druckauftrags als Feedback an die Drittanwendung zurücksenden wollen, sollten Sie diese Option nicht verwenden. Bleibt die Verbindung offen, können Sie mithilfe der Aktion **Daten an TCP/IP-Port senden** und dem Parameter *Absender antworten ein Feedback senden*.

- **Beim Empfang der Anzahl von Zeichen:** Legt fest, dass der Trigger auslöst, wenn die erforderliche Anzahl von Zeichen empfangen wird. In diesem Fall kann die Drittanwendung eine Verbindung offen lassen und kontinuierlich Daten senden. Jedes Datenpaket muss dieselbe Größe haben.
- **Beim Empfang der Abfolge von Zeichen:** Legt fest, dass der Trigger jedes Mal auslöst, wenn die erforderliche Anzahl von Zeichen empfangen wird. Diese Option wird verwendet, wenn Sie wissen, dass das Datenende immer durch eine identische Zeichenfolge gebildet wird. Anhand der Schaltfläche neben dem Feld „Bearbeiten“ können Sie (binäre) Sonderzeichen einfügen.

- **In Trigger-Daten einschließen:** Die Abfolge von Zeichen, die das Trigger-Ereignis bestimmt, wird nicht aus den Daten entfernt, sondern ist in den Daten enthalten. Der Trigger bezieht den vollständigen empfangenen Datenstrom.

- **Wenn nach dem angegebenen Zeitintervall nichts empfangen wird:** Gibt an, dass der Trigger auslöst, wenn ein bestimmtes Zeitintervall nach Empfang des letzten Zeichens verstreicht.

Ausführen

- **Verbindungen von den folgenden Hosts zulassen:** Gibt die Liste von IP-Adressen oder Hostnamen der Computer an, die sich mit dem Trigger verbinden dürfen. Stellen Sie jeden Eintrag in eine neue Zeile.
- **Verbindungen von den folgenden Hosts nicht zulassen:** Gibt die Liste von IP-Adressen oder Hostnamen der Computer an, die sich nicht mit dem Trigger verbinden dürfen. Stellen Sie jeden Eintrag in eine neue Zeile.
- **Begrüßungsnachricht:** Gibt eine Textnachricht an, die jedes Mal an den Client zurückgegeben wird, wenn er sich mit dem TCP/IP-Trigger verbindet.
- **Antwortmeldung:** Gibt die Textnachricht an, die jedes Mal an den Client zurückgegeben wird, wenn die Aktionen ausgeführt werden. Verwenden Sie diese Option, wenn der Client die Verbindung nicht nach Senden der Daten abbricht und die Antwort über das Ende der Aktionsausführung erwartet. Die Antwortnachricht ist fest codiert und daher immer identisch.
- **Meldungscodierung:** Gibt das Datencodierungsmuster an, damit Sonderzeichen richtig verarbeitet werden können.NiceLabel Automation kann die Datencodierung anhand des BOM-Headers (Textdateien) oder des Codierungs-Attributs (XML-Dateien) automatisch erkennen.

Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikationsparameter fest, durch die Sie Feedback von der Druck-Engine erhalten können.

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

- **Überwachtes Drucken:** Aktiviert den synchronen Druckmodus. Verwenden Sie ihn, wenn Sie den Status des Druckauftrags an die Drittanwendung zurücksenden wollen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Die Optionen im Abschnitt **Datenverarbeitung** geben an, ob Sie die Daten kürzen möchten, damit sie in eine Variable passen, oder ob die fehlenden Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet Fehler und unterbricht den Druckprozess, wenn Sie versuchen, zu lange Werte in Etikettenvariablen zu speichern oder Werte für nicht vorhandene Etikettenvariablen anzugeben.

- **Übermäßig lange Variableninhalte ignorieren:** kürzt Datenwerte, die die Länge der Variable gemäß Definition im Etiketten-Designer überschreiten, um sie auf passende Länge zu bringen. Diese Option wird wirksam, wenn Sie Variablenwerte in Filtern oder aus Befehlsdateien festlegen und wenn Sie Werte von Triggervariablen den gleichnamigen Etikettenvariablen zuordnen.

Beispiel

Die Etikettenvariable akzeptiert maximal 5 Zeichen. Wenn diese Option aktiviert ist, wird jeder Wert über 5 Zeichen auf die ersten 5 Zeichen gekürzt. Wenn der Wert 1234567 lautet, ignoriert die 6. Und 7. Stelle.

- **Fehlende Etikettenvariablen ignorieren:** Beim Drucken mit Befehlsdateien (z. B. JOB-Dateien), werden alle Variablen ignoriert, die:
  - in der Befehlsdatei angegeben sind (anhand des SET Befehls)
  - nicht auf dem Etikett definiert sind

  Etwas Ähnliches geschieht, wenn Sie einen Zuweisungsbereich in einem Filter definieren, um alle Name-Wert-Paare zu extrahieren, Ihr Etikett jedoch weniger Variablen enthält.
  Wenn Sie Werte von nicht vorhandenen Etikettenvariablen einstellen, gibt einen Fehler aus. Wenn diese Option aktiviert ist, wird der Druck fortgesetzt.

Die Optionen im Abschnitt **Scripting** geben die Scripting-Möglichkeiten an.

- **Scripting-Sprache:** Wählt die Scripting-Sprache für den Trigger aus. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden die ausgewählte Scripting-Sprache.

Die Optionen im Abschnitt **Empfangene Daten speichern** legen die Befehle fest, die für vom Trigger empfangene Daten zur Verfügung stehen.

- **Vom Trigger empfangene Daten speichern in Datei:** Aktivieren Sie diese Option, um vom Trigger empfangene Daten zu speichern. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und den Dateinamen enthält.
- **Bei Fehler vom Trigger empfangene Daten speichern in Datei:** Aktivieren sie diese Option, um die vom Trigger empfangenen Daten nur dann zu speichern, wenn ein Fehler bei der Ausführung der Aktion auftritt. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu behalten, die das Problem hervorgerufen haben, um die Lösungsfindung zu erleichtern.

ANMERKUNG

Stellen Sie sicher, dass Sie die Unterstützung für Überwachtes Drucken aktivieren. Ist sie nicht aktiviert, kann Fehler bei der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

ANMERKUNG

speichert die empfangenen Daten in einer temporären Datei. Diese temporäre Datei wird sofort nach Abschluss der Trigger-Ausführung gelöscht. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

Sicherheit

- **Trigger sperren und verschlüsseln:** Aktiviert den Trigger-Schutz. Wenn Sie die Option aktivieren, wird der Trigger gesperrt und kann nicht mehr bearbeitet werden. Dadurch werden die Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

Keepalive-Signale senden

Ihr Netzwerk-Routing-System kann Ihre Verbindung geräuschlos trennen, wenn einige Minuten lang kein TCP/IP-Verkehr stattfindet. Um zu vermeiden, dass die Verbindung unterbrochen wird, können Sie in regelmäßigen Abständen Keepalive-Signale in NiceLabel Automation.

Aktivieren Sie das Senden von Keepalive-Signalen in Ihrer Datei `product.config`:

1. Navigieren Sie zum Systemordner.
   %PROGRAMDATA%\NiceLabel\NiceLabel 10
2. Erstellen Sie eine Sicherungskopie der Datei **`product.config`**.
3. Öffnen Sie **`product.config`** in einem Text-Editor. Die Datei hat eine XML-Struktur.
4. Fügen Sie die folgenden Zeilen hinzu:

```
<configuration>
   <IntegrationService>
     <KeepAliveTime>60000</KeepAliveTime>
     <KeepAliveInterval>10000</KeepAliveInterval>
   </IntegrationService>
</configuration>
```

ANMERKUNG

`KeepAliveTime` (in Millisekunden): Gibt an, wie lange die TCP-Socket-Verbindung inaktiv sein muss, bevor die Automatisierung ein Keepalive-Signal sendet und auf die Rücksendung der Bestätigungspakete wartet.

`KeepAliveInterval` (in Millisekunden): Gibt an, in welchen Abständen ein weiteres Keepalive-Paket gesendet werden soll, wenn der Host keine Bestätigungspakete zurücksendet.

Wenn der TCP/IP-Trigger das KeepAlive-Paket empfängt und das Bestätigungspaket zurücksendet, beginnt die `KeepAliveTime`-Zeitschaltuhr erneut.

Beide Werte (`KeepAliveTime` und `KeepAliveInterval`) sind zwingend erforderlich, um das Senden von Keepalive-Signalen zu ermöglichen, und müssen Werte haben, die größer sind als 0.

5. Speichern Sie die Datei `product.config`.
6. Starten Sie Ihr Automation service neu.

Aktivierte Keepalive-Signale halten Ihre Verbindung jetzt aktiv.

3.5. TCP/IP Client Trigger

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

Der TCP/IP Client Trigger wandelt Ihr Automation in einen Daten empfangenden Client um, der sich mit TCP/IP-Servern verbindet. Es gibt mehrere Geräte und Systeme, die die Rolle eines TCP/IP-Servers übernehmen: Sichtkontrollsysteme, Drucker, SPS, Scanner, Waagen usw. Automation kann sich mit ihnen verbinden und auf eingehende Daten warten. Nach Empfang einer bestimmten Anzahl von Zeichen, einer Zeichenfolge oder nach einer Zeitüberschreitung wird der TCP/IP-Client-Trigger ausgelöst und beginnt mit der Ausführung Ihrer Aktionen. Wenn die Verbindung fehlschlägt, ermöglicht Ihnen der Trigger, automatisch eine erneute Verbindung herzustellen.

Typische Nutzung: Sie drucken automatisch mehrere Arten von Verpackungsetiketten mithilfe eines Netzwerkdruckers. Es ist unerlässlich für Sie, zu wissen, wann Ihr Drucker den Druck einer Art von Etikett abgeschlossen hat, bevor Sie mit der zweiten Etikettenart fortfahren. Der TCP/IP Client Trigger ermöglicht Ihnen, permanente automatisierte Prüfungen der Druckerverfügbarkeit festzulegen. Beim Drucken der ersten Etikettenart prüft der TCP/IP Client Trigger den Druckstatus. Wenn der Drucker fertig ist, weist er Automation an, die zweite Etikettenart zum Drucken zu senden.

Allgemein

- **Name:** Gibt den eindeutigen Namen des Triggers an. Die Namen helfen Ihnen dabei, zwischen verschiedenen Triggern zu unterscheiden, wenn Sie sie in Automation Builder konfigurieren und später in Automation Manager ausführen.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Triggers zu beschreiben. Geben Sie für die Benutzer eine kurze Erklärung zur Funktionsweise des Triggers ein.
- **Zielserver:** Geben Sie den Ort (IP-Adresse oder Hostname) des TCP/IP-Servers an, mit dem Sie eine Verbindung herstellen möchten.
- **Anschluss:** Gibt die Portnummer des Hosts an, von dem Sie die eingehenden Daten empfangen werden. Stellen Sie sicher, dass auf der Server-Seite die entsprechenden Ports in der Firewall geöffnet sind.
- **Intervall für erneute Verbindung zum Server**: Geben Sie die Zeit in Millisekunden an, nach der Automation versucht, eine erneute Verbindung zu Ihrem TCP/IP-Server herzustellen.
- **Beim Empfang der Anzahl von Zeichen:** Legt fest, dass der Trigger auslöst, wenn die erforderliche Anzahl von Zeichen empfangen wird. In diesem Fall kann der Server eine Verbindung offen lassen und kontinuierlich Daten senden. Jedes Datenpaket muss dieselbe Größe haben.

- **Beim Empfang der Abfolge von Zeichen:** Legt fest, dass der Trigger jedes Mal auslöst, wenn die erforderliche Anzahl von Zeichen empfangen wird. Diese Option wird verwendet, wenn Sie wissen, dass das Datenende immer durch eine identische Zeichenfolge gebildet wird. Anhand der Schaltfläche neben dem Feld „Bearbeiten“ können Sie (binäre) Sonderzeichen einfügen.
    - **In Trigger-Daten einschließen:** Die Abfolge von Zeichen, die das Trigger-Ereignis bestimmt, wird nicht aus den Daten entfernt, sondern ist in den Daten enthalten. Der Trigger bezieht den vollständigen empfangenen Datenstrom.
- **Wenn nach dem angegebenen Zeitintervall nichts empfangen wird:** Gibt an, dass der Trigger auslöst, wenn ein bestimmtes Zeitintervall nach Empfang des letzten Zeichens verstreicht.

Ausführen

- **Initialisierungsnachricht**: Textnachricht, die jedes Mal an den Server gesendet wird, wenn Automation die Verbindung herstellt.
- **Antwortmeldung:** Gibt die Textnachricht an, die beim Auslösen des Triggers an den Server ausgegeben wird (bevor die Aktionen ausgeführt werden).
- **Meldungscodierung:** Gibt das Datencodierungsmuster an, damit Sonderzeichen richtig verarbeitet werden können.NiceLabel Automation kann die Datencodierung anhand des BOM-Headers (Textdateien) oder des Codierungs-Attributs (XML-Dateien) automatisch erkennen.

Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikationsparameter fest, durch die Sie Feedback von der Druck-Engine erhalten können.

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

- **Überwachtes Drucken:** Aktiviert den synchronen Druckmodus. Verwenden Sie ihn, wenn Sie den Status des Druckauftrags an die Drittanwendung zurücksenden wollen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Die Optionen im Abschnitt **Datenverarbeitung** geben an, ob Sie die Daten kürzen möchten, damit sie in eine Variable passen, oder ob die fehlenden Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet Fehler und unterbricht den Druckprozess, wenn Sie versuchen, zu lange Werte in Etikettenvariablen zu speichern oder Werte für nicht vorhandene Etikettenvariablen anzugeben.

- **Übermäßig lange Variableninhalte ignorieren:** kürzt Datenwerte, die die Länge der Variable gemäß Definition im Etiketten-Designer überschreiten, um sie auf passende Länge zu bringen. Diese Option wird wirksam, wenn Sie Variablenwerte in Filtern oder aus Befehlsdateien festlegen und wenn Sie Werte von Triggervariablen den gleichnamigen Etikettenvariablen zuordnen.

Beispiel

Die Etikettenvariable akzeptiert maximal 5 Zeichen. Wenn diese Option aktiviert ist, wird jeder Wert über 5 Zeichen auf die ersten 5 Zeichen gekürzt. Wenn der Wert 1234567 lautet, ignoriert die 6. Und 7. Stelle.

- **Fehlende Etikettenvariablen ignorieren:** Beim Drucken mit Befehlsdateien (z. B. JOB-Dateien), werden alle Variablen ignoriert, die:
  - in der Befehlsdatei angegeben sind (anhand des SET Befehls)
  - nicht auf dem Etikett definiert sind

  Etwas Ähnliches geschieht, wenn Sie einen Zuweisungsbereich in einem Filter definieren, um alle Name-Wert-Paare zu extrahieren, Ihr Etikett jedoch weniger Variablen enthält.
  Wenn Sie Werte von nicht vorhandenen Etikettenvariablen einstellen, gibt einen Fehler aus. Wenn diese Option aktiviert ist, wird der Druck fortgesetzt.

Die Optionen im Abschnitt **Scripting** geben die Scripting-Möglichkeiten an.

- **Scripting-Sprache:** Wählt die Scripting-Sprache für den Trigger aus. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden die ausgewählte Scripting-Sprache.

Die Optionen im Abschnitt **Empfangene Daten speichern** legen die Befehle fest, die für vom Trigger empfangene Daten zur Verfügung stehen.

- **Vom Trigger empfangene Daten speichern in Datei:** Aktivieren Sie diese Option, um vom Trigger empfangene Daten zu speichern. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und den Dateinamen enthält.
- **Bei Fehler vom Trigger empfangene Daten speichern in Datei:** Aktivieren sie diese Option, um die vom Trigger empfangenen Daten nur dann zu speichern, wenn ein Fehler bei der Ausführung der Aktion auftritt. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu behalten, die das Problem hervorgerufen haben, um die Lösungsfindung zu erleichtern.

ANMERKUNG

Stellen Sie sicher, dass Sie die Unterstützung für Überwachtes Drucken aktivieren. Ist sie nicht aktiviert, kann Fehler bei der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

ANMERKUNG

speichert die empfangenen Daten in einer temporären Datei. Diese temporäre Datei wird sofort nach Abschluss der Trigger-Ausführung gelöscht. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

Sicherheit

- **Trigger sperren und verschlüsseln:** Aktiviert den Trigger-Schutz. Wenn Sie die Option aktivieren, wird der Trigger gesperrt und kann nicht mehr bearbeitet werden. Dadurch werden die Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

Keepalive-Signale senden

Ihr Netzwerk-Routing-System kann Ihre Verbindung geräuschlos trennen, wenn einige Minuten lang kein TCP/IP-Verkehr stattfindet. Um zu vermeiden, dass die Verbindung unterbrochen wird, können Sie in regelmäßigen Abständen Keepalive-Signale in NiceLabel Automation.

Aktivieren Sie das Senden von Keepalive-Signalen in Ihrer Datei `product.config`:

1. Navigieren Sie zum Systemordner.
   %PROGRAMDATA%\NiceLabel\NiceLabel 10
2. Erstellen Sie eine Sicherungskopie der Datei **`product.config`**.
3. Öffnen Sie **`product.config`** in einem Text-Editor. Die Datei hat eine XML-Struktur.
4. Fügen Sie die folgenden Zeilen hinzu:

```
<configuration>
   <IntegrationService>
     <KeepAliveTime>60000</KeepAliveTime>
     <KeepAliveInterval>10000</KeepAliveInterval>
   </IntegrationService>
</configuration>
```

ANMERKUNG

`KeepAliveTime` (in Millisekunden): Gibt an, wie lange die TCP-Socket-Verbindung inaktiv sein muss, bevor die Automatisierung ein Keepalive-Signal sendet und auf die Rücksendung der Bestätigungspakete wartet.

`KeepAliveInterval` (in Millisekunden): Gibt an, in welchen Abständen ein weiteres Keepalive-Paket gesendet werden soll, wenn der Host keine Bestätigungspakete zurücksendet.

Wenn der TCP/IP-Trigger das KeepAlive-Paket empfängt und das Bestätigungspaket zurücksendet, beginnt die `KeepAliveTime`-Zeitschaltuhr erneut.

Beide Werte (`KeepAliveTime` und `KeepAliveInterval`) sind zwingend erforderlich, um das Senden von Keepalive-Signalen zu ermöglichen, und müssen Werte haben, die größer sind als 0.

5. Speichern Sie die Datei `product.config`.
6. Starten Sie Ihr Automation service neu.

Aktivierte Keepalive-Signale halten Ihre Verbindung jetzt aktiv.

3.6. HTTP Server Trigger

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

Der HTTP Server Trigger löst aus, wenn Daten am überwachten Socket (IP-Adresse und Portnummer) empfangen werden.

Im Gegensatz zum TCP/IP-Trigger werden diese Daten nicht als Rohdatenstrom übermittelt, sondern enthalten den Standard-HTTP-Header. Drittanwendungen müssen POST- oder GET-Anfragemethoden nutzen. Sie müssen Daten im Nachrichtenkörper oder in der Abfrage-Zeichenfolge bereitstellen. Sie können im Nachrichtenkörper beide Internet-Medientypen verwenden – MIME-Typ oder Content-Typ.NiceLabel Automation empfängt die Nachricht und extrahiert relevante Daten anhand eines Filters aus dem Nachrichteninhalt.

Typische Nutzung: Das vorhandene Geschäftssystem führt eine Transaktion durch, wodurch als HTTP POST-Nachricht formatierte Daten an ein bestimmtes Socket des NiceLabel Automation Servers gesendet werden. Die gesendeten Daten können strukturiert sein (CSV, XML und andere Formate) oder ein proprietäres veraltetes Format aufweisen. In jedem Fall liest NiceLabel Automation die Daten aus, parst Werte anhand von Filtern und druckt die extrahierten Daten auf Etiketten. Weitere Informationen zum Parsen und Extrahieren von Daten finden Sie im Abschnitt Informationen zu Filtern.

TIPP

Hilfe beim Erstellen von Konfigurationen mit dem HTTP Server Trigger finden Sie in der Automation Beispieldatei Label Preview as HTTP Response. Sie finden Beispieldateien unter **Hilfe** > **Beispieldateien**.

Daten bereitstellen

Stellen Sie die HTTP-Triggerdaten anhand einer der folgenden Methoden bereit. Sie können die Methoden bei Bedarf auch innerhalb derselben HTTP-Abfrage kombinieren.

Daten in der Abfragefolge

Eine Abfragefolge ist Teil eines Uniform Resource Locators (URL), der Daten enthält, welche an den HTTP-Trigger übermittelt werden sollen.

Beispiel für eine typische URL mit Abfragefolge:

```
http://server/path/?query_string
```

Das Fragezeichen wird als Trennzeichen verwendet und ist nicht Teil der Abfragefolge.

Eine Abfragefolge besteht normalerweise aus einer Reihe von **`Name:Wert`**-Paaren. In jedem Paar sind Feldname und Wert durch ein Gleichheitszeichen (=) getrennt. Reihen von Paaren werden jeweils durch ein Et-Zeichen (&) getrennt. Eine typische Abfragefolge stellt Werte für Felder (Variablen) im folgenden Format bereit:

```
field1=value1&field2=value2&field3=value3
```

Der HTTP-Trigger bietet integrierte Unterstützung für die Extraktion von Werten aus allen Feldern und deren Speichern in Variablen mit identischem Namen. Daher müssen Sie keine Filter definieren, um Werte aus der Abfragefolge zu extrahieren.

- Sie müssen keine Variablen innerhalb eines Triggers definieren, um diese mit Werten aus der Abfragefolge zu füllen.NiceLabel Automation extrahiert alle Variablen aus der Abfragefolge und sendet ihre Werte an das aktuell aktive Etikett. Falls Variablen mit identischen Namen im Etikett vorhanden sind, füllt Automation sie mit Werten. Wenn die Variablen nicht im Etikett vorhanden sind, ignoriert Automation ihre Werte, ohne Fehler zu melden.
- Wenn eine Aktion Variablenwerte erfordert, legen Sie entsprechende Variablen im Trigger an. Um alle Werte aus der Abfragefolge abzurufen und zu speichern, erstellen Sie entsprechende Variablen, deren Namen mit denen der Felder in der Abfragefolge identisch sind. Im obigen Beispiel würden Sie also Triggervariablen mit den Namen **`feld1`**, **`feld2`** und **`feld3`** definieren.

Normalerweise verwenden Sie die GET HTTP-Abfragemethode, um die Abfragefolge bereitzustellen.

Daten im Textkörper einer HTTP-Anfrage

Verwenden Sie die POST-Abfragemethode, um eine Nachricht im Textkörper einer HTTP-Anfrage bereitzustellen.

Sie können alle Arten von Daten senden und jede gewünschte Datenstruktur im Textkörper verwenden. Sie müssen nur sicherstellen, dass Sie die Daten anhand von NiceLabel Automation Filtern handhaben können. Der Inhalt kann als XML, CSV oder reiner Text formatiert sein. Gesendeter Inhalt kann sogar im Binärdatenformat (Base64-codiert) vorliegen. Denken Sie daran, dass die Daten mit Filtern geparst werden müssen.

Falls Sie Einfluss auf die Struktur der eingehenden Nachricht haben, sollten Sie standardisierte Strukturen wie XML oder CSV verwenden, um die Filterkonfiguration zu vereinfachen.

Um die Daten im Textkörper bereitzustellen, verwenden Sie die POST HTTP-Abfragemethode.

Allgemein

In diesem Bereich können Sie die wichtigsten Dateitrigger-Einstellungen vornehmen.

- **Name:** Gibt den eindeutigen Namen des Triggers an. Die Namen helfen Ihnen dabei, zwischen verschiedenen Triggern zu unterscheiden, wenn Sie sie in Automation Builder konfigurieren und später in Automation Manager ausführen.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Triggers zu beschreiben. Geben Sie für die Benutzer eine kurze Erklärung zur Funktionsweise des Triggers ein.

Kommunikation

ANMERKUNG

Dieser Trigger unterstützt Internet Protocol Version 6 (IPv6).

In diesem Bereich können Sie die obligatorische Portnummer sowie verschiedene Feedback-Optionen konfigurieren. Sie können Standard-HTTP-Antwortcodes verwenden, um eine erfolgreich ausgeführte Aktion anzuzeigen. Für spezifischere Zwecke können Sie individuellen Inhalt auch zurück an die datengebende Anwendung senden. Bei solchem Inhalt kann es sich um eine einfache Feedback-Zeichenfolge oder um Binärdaten wie eine Etikettenvorschau oder einen Druckstrom handeln.

Die typische URL zur Verbindung mit einem HTTP-Trigger sieht folgendermaßen aus:

```
http://server:port/path/?query_string
```

- **Server:** Dies ist der FQDN oder die IP-Adresse des Rechners auf dem NiceLabel Automation installiert ist.
- **Anschluss:** Nummer der Schnittstelle, an dem die eingehenden Daten empfangen werden. Verwenden Sie eine Schnittstelle, die noch nicht von einer anderen Anwendung genutzt wird. Falls die ausgewählte Schnittstelle in Verwendung ist, können Sie den Trigger nicht in Automation Manager starten. Weitere Informationen zu Sicherheitsüberlegungen finden Sie in Abschnitt Zugriff auf Ihre Trigger sichern.

  ANMERKUNG

  Wenn auf Ihrem Server Multi-Homing aktiviert ist (mehrere IP-Adressen auf einer oder mehreren Netzwerkkarten), antwortet NiceLabel Automation an der festgelegten Schnittstelle für alle IP-Adressen.

- **Path:** Gibt den optionalen Pfad in der URL an. Diese Funktion ermöglicht es NiceLabel Automation, mehrere HTTP-Trigger an derselben Schnittstelle anzubieten. Der Client nutzt die Trigger über eine einzelne Schnittstelle mit einer REST-ähnlichen Syntax, wodurch verschiedene Trigger durch andere URLs ausgelöst werden. Wenn Sie nicht wissen, welchen Pfad Sie verwenden sollen, behalten Sie die Standardeinstellung (\) bei.

> **PRODUKTEBENEN-INFO**
> Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

- **Sichere Verbindung (HTTPS):** Aktiviert die sichere Transportschicht für Ihre HTTP-Nachrichten und verhindert Abhörangriffe. Weitere Informationen zur Einrichtung finden Sie im Abschnitt Sicheres Übertragungsprotokoll (HTTPS) nutzen.
- **Abfragefolge:** Gibt die Name-Wert-Paare in der URL an. Dies ist ein optionaler Parameter. Die Daten werden normalerweise im Textkörper der HTTP-Anfrage bereitgestellt.
- **Warten, bis Trigger-Ausführung abgeschlossen ist:** Das HTTP-Protokoll wartet darauf, dass der Empfänger (in diesem Fall NiceLabel Automation) eine numerische Antwort an den Absender übermittelt, die den Status der empfangenen Nachricht anzeigt. Standardmäßig antwortet NiceLabel Automation mit dem Code 200. Dies zeigt an, dass Automation die Daten erfolgreich empfangen hat, gibt aber keine Informationen über den Erfolg von Trigger-Aktionen.
  Diese Option legt fest, dass ein Trigger nicht sofort nach Empfang der Daten eine Antwort sendet, sondern wartet, bis alle Aktionen ausgeführt wurden. Danach sendet er den Antwort-Code, um das erfolgreiche Ausführen von Aktionen zu bestätigen. Wenn diese Option aktiviert ist, können Sie eine benutzerdefinierte Antwort und die entsprechenden Daten zurücksenden (z. B. ist die Antwort auf eine HTTP-Abfrage die Etikettenvorschau im PDF-Format).
  Die verfügbaren integrierten HTTP-Antwortcodes sind:

HTTP-Antwortcode	Beschreibung
200	Alle Aktionen erfolgreich ausgeführt.
401	Nicht autorisiert, falscher Benutzername und Passwort angegeben.
500	Beim Ausführen der Aktion sind Fehler aufgetreten.

> **ANMERKUNG**
> Um Feedback zum Druckprozess zu senden, aktivieren Sie den **synchronen** Druckmodus. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

- **Maximale Anzahl gleichzeitiger Anfragen:** Gibt die maximale Anzahl gleichzeitig akzeptierter Verbindungen an. Dies definiert die Anzahl von Clients, die gleichzeitig Daten an den Trigger senden können. Diese Anzahl hängt auch von der Hardwareleistung Ihres Servers ab.
  Mehr darüber erfahren Abschnitt 7.1, „Parallele Verarbeitung“.
- **Antworttyp:** Gibt den Typ Ihrer Antwortnachricht an. Häufig verwendete Internet-Medientypen (auch MIME-Typen oder Content-Typen genannt) stehen in der Dropdown-Liste zur Verfügung. Falls Ihr Medientyp nicht in der Liste enthalten ist, geben Sie ihn selbst ein. Automation sendet die ausgehenden Antwortdaten im Format des festgelegten Medientyps als Feedback. **Variable** aktiviert variable Medientypen. Wählen Sie bei aktivierter Option eine Variable aus (oder erstellen Sie eine Variable), die den Medientyp enthält.

ANMERKUNG

Wenn Sie keinen MIME-Typ angeben, verwendet NiceLabel Automation standardmäßig **`application/octet-stream`**.

- **Antwortdaten:** Definiert den Inhalt der Antwortnachricht. Beispiele für Inhalte, die Sie in Form von HTTP-Antworten senden können: Benutzerdefinierte Fehlermeldungen, eine Etikettenvorschau, erzeugte PDF-Dateien, Druckstrom-(Spool-)Dateien, XML-Dateien mit Daten aus der Druck-Engine und der Etikettenvorschau (als Base64-Zeichenfolge codiert) usw. Die Möglichkeiten sind praktisch unbegrenzt.
  Wenn Ihre Ausgabe ausschließlich aus binären Inhalten besteht (z. B. Etikettenvorschau oder Druckstrom), müssen Sie den korrekten Medientyp auswählen, z. B. **`image/jpeg`** oder **`application/octet-stream`**.

- **Zusätzliche Header**: Ermöglicht es Ihnen, benutzerdefinierte MIME-Header für die HTTP-Antwortnachricht zu definieren.
  Die Syntax für Antwortheader und ein Beispiel stehen im Abschnitt Aktion HTTP-Anfrage zur Verfügung.

TIPP

Sie können bei Antwortdaten und zusätzlichen Headern festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzufügen, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie die Variable aus der Liste ein (oder erstellen Sie eine neue Variable), welche die Daten enthält, die Sie verwenden möchten. Weitere Informationen finden Sie im Abschnitt Zusammengesetzte Werte verwenden im Benutzerhandbuch.

Authentifizierung

- **Keine:** Es wird keine Authentifizierungsmethode verwendet.
- **Benutzer:** Gibt an, dass eingehende Nachrichten einen Benutzernamen und ein Passwort enthalten. Wenn diese Option verwendet wird, akzeptiert der Trigger nur HTTP-Nachrichten mit übereinstimmenden Zugangsdaten. Weitere Informationen zu Sicherheitsüberlegungen finden Sie in Abschnitt Zugriff auf Ihre Trigger sichern.
- **Anwendungsgruppe (definiert in NiceLabel Control Center):** Ebenso wie beim **Benutzer**-Authentifizierungstyp legt diese Option zudem fest, dass die eingehenden Nachrichten Benutzernamen und Passwort enthalten. Wenn diese Option verwendet wird, akzeptiert der Trigger nur HTTP-Nachrichten mit entsprechenden Anmeldedaten für NiceLabel Control Center Benutzer, die einer bestimmen Anwendungsgruppe angehören.
  - **Gruppe**: Mehrere Anwendungsgruppen können im NiceLabel Control Center definiert werden. Um auszuwählen, welche Gruppe auf den HTTP-Server Trigger zugreifen darf, verwenden Sie die Dropdown-Liste **Gruppe**. Die ausgewählte Gruppe muss, ebenso wie ihre Benutzer, als aktiv definiert sein, wenn der Trigger ausgeführt wird.

> **ANMERKUNG**
> Eine Gruppe mit einem festgelegten Namen muss im NiceLabel Control Center vorhanden sein, wenn der Trigger ausgeführt wird. Bei der Arbeit an der Konfiguration im Automation Builder können Sie einen beliebigen Gruppennamen verwenden. Definieren Sie später im NiceLabel Control Center einen finalen Namen und übernehmen Sie diesen in die Konfiguration, bevor Sie sie implementieren.

> **TIPP**
> Die Benutzer authentifizieren sich selbst anhand ihrer Anmeldedaten, wie unter **NiceLabel Control Center** > **Verwaltung** > **Benutzer und Gruppen**. Im NiceLabel Control Center Benutzerhandbuch finden Sie weitere Informationen zur Benutzerverwaltung (Kapitel „Benutzer und Gruppen“).

Sonstiges
Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikationsparameter fest, durch die Sie Feedback von der Druck-Engine erhalten können.

> **PRODUKTEBENEN-INFO**
> Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

- **Überwachtes Drucken:** Aktiviert den synchronen Druckmodus. Verwenden Sie ihn, wenn Sie den Status des Druckauftrags an die Drittanwendung zurücksenden wollen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Die Optionen im Abschnitt **Datenverarbeitung** geben an, ob Sie die Daten kürzen möchten, damit sie in eine Variable passen, oder ob die fehlenden Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet Fehler und unterbricht den Druckprozess, wenn Sie versuchen, zu lange Werte in Etikettenvariablen zu speichern oder Werte für nicht vorhandene Etikettenvariablen anzugeben.

- **Übermäßig lange Variableninhalte ignorieren:** kürzt Datenwerte, die die Länge der Variable gemäß Definition im Etiketten-Designer überschreiten, um sie auf passende Länge zu bringen. Diese Option wird wirksam, wenn Sie Variablenwerte in Filtern oder aus Befehlsdateien festlegen und wenn Sie Werte von Triggervariablen den gleichnamigen Etikettenvariablen zuordnen.

> **Beispiel**
> Die Etikettenvariable akzeptiert maximal 5 Zeichen. Wenn diese Option aktiviert ist, wird jeder Wert über 5 Zeichen auf die ersten 5 Zeichen gekürzt. Wenn der Wert 1234567 lautet, ignoriert die 6. Und 7. Stelle.

- **Fehlende Etikettenvariablen ignorieren:** Beim Drucken mit Befehlsdateien (z. B. JOB-Dateien), werden alle Variablen ignoriert, die:

- in der Befehlsdatei angegeben sind (anhand des SET Befehls)
- nicht auf dem Etikett definiert sind

Etwas Ähnliches geschieht, wenn Sie einen Zuweisungsbereich in einem Filter definieren, um alle Name-Wert-Paare zu extrahieren, Ihr Etikett jedoch weniger Variablen enthält.
Wenn Sie Werte von nicht vorhandenen Etikettenvariablen einstellen, gibt einen Fehler aus. Wenn diese Option aktiviert ist, wird der Druck fortgesetzt.

Die Optionen im Abschnitt **Scripting** geben die Scripting-Möglichkeiten an.

- **Scripting-Sprache:** Wählt die Scripting-Sprache für den Trigger aus. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden die ausgewählte Scripting-Sprache.

Die Optionen im Abschnitt **Empfangene Daten speichern** legen die Befehle fest, die für vom Trigger empfangene Daten zur Verfügung stehen.

- **Vom Trigger empfangene Daten speichern in Datei:** Aktivieren Sie diese Option, um vom Trigger empfangene Daten zu speichern. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und den Dateinamen enthält.
- **Bei Fehler vom Trigger empfangene Daten speichern in Datei:** Aktivieren sie diese Option, um die vom Trigger empfangenen Daten nur dann zu speichern, wenn ein Fehler bei der Ausführung der Aktion auftritt. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu behalten, die das Problem hervorgerufen haben, um die Lösungsfindung zu erleichtern.

ANMERKUNG

Stellen Sie sicher, dass Sie die Unterstützung für Überwachtes Drucken aktivieren. Ist sie nicht aktiviert, kann Fehler bei der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

ANMERKUNG

speichert die empfangenen Daten in einer temporären Datei. Diese temporäre Datei wird sofort nach Abschluss der Trigger-Ausführung gelöscht. Die interne Variable **`DataFileName`** verweist auf diesen Dateinamen. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

Sicherheit

- **Trigger sperren und verschlüsseln:** Aktiviert den Trigger-Schutz. Wenn Sie die Option aktivieren, wird der Trigger gesperrt und kann nicht mehr bearbeitet werden. Dadurch werden die Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

3.7. Webdienst-Trigger

PRODUKTEBENEN-INFO

Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

Das Webdienst-Triggerereignis tritt ein, wenn ein überwachtes Socket (IP-Adresse und Portnummer) Daten empfängt. Die Daten müssen der SOAP-Notation entsprechen – sie codiert XML-Daten als HTTP-Nachricht. Die Webdienst-Schnittstelle wird im WSDL-Dokument beschrieben. Ein solches Dokument steht für jeden definierten Webdienst-Trigger zur Verfügung.

Der Webdienst-Trigger stellt ein Feedback zum Druckauftrag bereit, aber Sie müssen dazu den **synchronen** Verarbeitungsmodus aktivieren. Weitere Informationen finden Sie im Abschnitt Feedback zum Status von Druckaufträgen im Benutzerhandbuch.

Normalerweise verwenden Programmierer den Webdienst, um den Etikettendruck in ihre eigenen Anwendungen zu integrieren. Ein vorhandenes Geschäftssystem führt eine Transaktion durch, wodurch Daten über ein bestimmtes Socket an den NiceLabel Automation Server gesendet werden. Die gesendeten Daten sind als SOAP-Nachricht formatiert. Die Daten können in einem strukturierten Format wie CSV und XML oder aber in einem der Legacy-Formate bereitgestellt werden. In beiden Fällen liest NiceLabel Automation die Daten aus, parst Werte anhand von Filtern und druckt diese Werte auf Etiketten. Weitere Informationen zum Parsen und Extrahieren von Daten finden Sie im Abschnitt Informationen zu Filtern.

TIPP

Hilfe beim Erstellen von Konfigurationen mit dem Webdienst-Trigger finden Sie in der Automation Beispieldatei Web Service. Sie finden Beispieldateien unter **Hilfe** > **Beispieldateien**.

Allgemein

In diesem Bereich können Sie die allgemeinen Datei-Trigger-Einstellungen vornehmen.

- **Name:** Gibt den eindeutigen Namen des Triggers an. Die Namen helfen Ihnen dabei, zwischen verschiedenen Triggern zu unterscheiden, wenn Sie sie in Automation Builder konfigurieren und später in Automation Manager ausführen.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Triggers zu beschreiben. Geben Sie für die Benutzer eine kurze Erklärung zur Funktionsweise des Triggers ein.

Kommunikation

ANMERKUNG

Dieser Trigger unterstützt Internet Protocol Version 6 (IPv6).

In diesem Bereich können Sie die obligatorische Portnummer sowie verschiedene Feedback-Einstellungen konfigurieren.

- **Anschluss:** Gibt die Nummer der Schnittstelle an, die eingehende Daten empfängt. Verwenden Sie eine Schnittstelle, die noch nicht von einer anderen Anwendung genutzt wird. Falls die ausgewählte Schnittstelle in Verwendung ist, können Sie den Trigger nicht in Automation Manager starten. Weitere Informationen zu Sicherheitsüberlegungen finden Sie in Abschnitt Zugriff auf Ihre Trigger sichern.

  ANMERKUNG

  Wenn auf Ihrem Server Multi-Homing aktiviert ist (mehrere IP-Adressen auf einer oder mehreren Netzwerkkarten), antwortet NiceLabel Automation an der festgelegten Schnittstelle für alle IP-Adressen.

- **Sichere Verbindung (HTTPS):** Aktiviert die sichere Transportschicht für Ihre HTTP-Nachrichten und verhindert Abhörangriffe. Weitere Informationen zur Einrichtung finden Sie im Abschnitt Sicheres Übertragungsprotokoll (HTTPS) nutzen.
- **Maximale Anzahl gleichzeitiger Aufrufe:** Gibt die maximale Anzahl gleichzeitig akzeptierter Verbindungen an. Dies definiert die Anzahl von Clients, die gleichzeitig Daten an den Trigger senden können.
- **Antwortdaten:** Definiert die benutzerdefinierte Antwort, die zusammen mit den Methoden `ExecuteTriggerWithResponse` und `ExecuteTriggerAndSetVariablesWithResponse` verwendet werden kann. Die Antwortdaten enthalten den Inhalt, der im Textbereich angezeigt wird. Sie können feste Werte, variable Werte und Sonderzeichen verbinden. Um Variablen und Sonderzeichen einzufügen (oder zu erstellen), klicken Sie auf die Pfeil-Schaltfläche rechts neben dem Textbereich. Die Antwort kann Binärdaten enthalten, z. B. Etikettenvorschaubilder und Druckdateien (*.PRN).

Status-Feedback

Standardmäßig stellt der Trigger Feedback zum Status des erstellten Druckauftrags bereit. Der Trigger akzeptiert und nutzt die bereitgestellten Daten zum Ausführen definierter Aktionen. Sie können die Ausführung von Aktionen überwachen – der Trigger meldet einen Erfolgsstatus für jedes Ereignis, das während der Ausführung eintritt. Um Statusberichte während des Druckvorgangs zu ermöglichen, aktivieren Sie Synchroner Druckmodus.

Der Webdienst-Trigger stellt die folgenden Methoden (Funktionen) zur Verfügung:

- **ExecuteTrigger:** Diese Methode nimmt Daten zur Verarbeitung entgegen und gibt optionales Statusfeedback aus. Einer der Eingabeparameter aktiviert bzw. deaktiviert Feedback. Wenn Sie Statusberichte aktivieren, enthält das Feedback die Fehler-ID und eine ausführliche Beschreibung des Fehlers. Ist die Fehler-ID gleich 0, gab es kein Problem beim Erstellen der Druckdatei. Ist die Fehler-ID größer als 0, ist beim Druckprozess ein Fehler aufgetreten. Die Webdienst-Antwort in dieser Methode ist nicht konfigurierbar – die Antwort enthält immer die Fehler-ID und die Fehlerbeschreibung.

- **ExecuteTriggerWithResponse:** Diese Methode nimmt Daten zur Verarbeitung entgegen und gibt benutzerdefiniertes Statusfeedback aus. Die Webdienst-Antwort ist konfigurierbar. Sie können mit allen Arten von Daten antworten, die nach einer beliebigen verfügbaren Struktur organisiert sind. Sie können binäre Daten in der Antwort verwenden.
- **ExecuteTriggerAndSetVariables:** Ähnlich wie die obige Methode **ExecuteTrigger**, bietet dieser Webdienst einen zusätzlichen eingehenden Parameter, welcher die formatierte Liste mit *Name-Wert*-Paaren entgegennimmt. Der Trigger parst die Liste automatisch, extrahiert Werte und speichert die Werte in den gleichnamigen Variablen, sodass Sie keinen eigenen Filter für die Datenextraktion definieren müssen.
- **ExecuteTriggerAndSetVariablesWithResponse:** Ähnlich wie die obige Methode **ExecuteTriggerWithResponse**, bietet dieser Webdienst einen zusätzlichen eingehenden Parameter, welcher die formatierte Liste mit *Name-Wert*-Paaren entgegennehmen. Der Trigger parst die Liste automatisch, extrahiert Werte und speichert die Werte in der gleichnamigen Variablen, sodass Sie keinen eigenen Filter für die Datenextraktion definieren müssen.

Weitere Informationen über die Struktur von Nachrichten, die Sie anhand einer der oben aufgeführten Methoden senden können, finden Sie im Abschnitt WSDL unten.

WSDL
Die Web Services Description Language (WSDL) gibt den Stil von SOAP-Nachrichten vor. Dies kann entweder ein **Remoteprozeduraufruf-Stil (RPC)** oder ein **Dokument-Stil** sein. Wählen Sie den Stil, den Ihre datengebende Anwendung unterstützt.

Das WSDL-Dokument legt die Eingangs- und Ausgangsparameter des Webdienstes fest.

Nachdem der Webdienst-Trigger an Port 12345 definiert wurde, implementieren Sie ihn in Automation Manager und starten Sie ihn. WSDL wird verfügbar unter:

```
http://localhost:12345
```

WSDL bietet verschiedene Methoden an, die allesamt Daten für den Trigger bereitstellen. Wählen Sie die Methode, die für Ihre Anforderungen am besten geeignet ist.

- Methoden mit *WithResponse* im Namen ermöglichen es Ihnen, angepasste Antworten zu senden, etwa benutzerdefinierte Fehlermeldungen, eine Etikettenvorschau, PDF-Dateien, Druckdateien (*.PRN) und ähnliches. Methoden ohne *WithResponse* im Namen geben ebenfalls Feedback aus, aber Sie können die Antwort nicht anpassen. Das Feedback enthält standardmäßige Fehlermeldungen.
- Methoden mit *SetVariables* im Namen ermöglichen Ihnen, eine Liste mit Variablen in zwei vordefinierten Formaten bereitzustellen.Automation Extrahiert automatisch Werte und ordnet sie den passenden Variablen zu. Dies spart Ihnen Zeit, weil Sie keinen Filter für die Extraktion und Zuordnung definieren müssen. Für die Methoden ohne *SetVariables* im Namen müssen Sie die Filter selbst definieren.

Die Oberfläche des Webdienstes definiert die folgenden Methoden:

ExecuteTrigger-Methode
Der Hauptteil der Definition lautet:

```
<wsdl:message name="WebSrviTrg_ExecuteTrigger_InputMessage">
    <wsdl:part name="text" type="xsd:string"/>
    <wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="WebSrviTrg_ExecuteTrigger_OutputMessage"
    <wsdl:part name="ExecuteTriggerResult" type="xsd:int"/
    <wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Die Definition beinhaltet zwei Eingabevariablen (Sie geben ihre Werte an):

- **text:** Filter in der Konfiguration parst diese Eingabezeichenfolge. Normalerweise ist die Eingabezeichenfolge als CSV- oder XML-Datei strukturiert, wodurch sie sich einfach parsen lässt. Sie können auch ein anderes Textdateiformat verwenden.
- **wait:** Dies ist ein boolesches Feld, das zwei Dinge festlegt:
  - Ob Sie auf die Statusantwort des Druckauftrags warten möchten oder nicht.
  - Ob der Webdienst ein Feedback ausgeben soll oder nicht.

  Wählen Sie für *Wahr* den Wert **1**. Wählen Sie für *Falsch* den Wert **0**. Abhängig vom ausgewählten Methodentyp gibt es entweder eine vordefinierte Antwort, oder Sie können eine benutzerdefinierte Antwort senden.

Dies sind optionalen Ausgabevariablen (Sie erhalten ihre Werte, wenn Sie sie anfordern, indem Sie **wait** auf **1** setzen:

- **ExecuteTriggerResult:** Die als Ganzzahl ausgegebene Antwort enthält den Wert 0, falls kein(e) Datenverarbeitungsfehler gemeldet wird/werden. Sie enthält eine Ganzzahl größer 0, wenn Fehler aufgetreten sind. Die Anwendung, welche den Webdienst-Aufruf an NiceLabel Automation ausführt, kann die Antwort als Fehlerindikator verwenden.
- **errorText:** Dieser Zeichenfolgenwert enthält eine Druckauftrags-Statusantwort, wenn ein Fehler während der Trigger-Verarbeitung auftritt.

  > **ANMERKUNG**
  > Wenn während der Trigger-Verarbeitung ein Fehler auftritt, wird dieses Element in die XML-Antwortnachricht eingeschlossen und sein Wert enthält die Fehlerbeschreibung. Ist jedoch kein Fehler aufgetreten, wird das Element nicht in die XML-Antwort eingeschlossen.

ExecuteTriggerWithResponse-Methode
Sie würden diese Methode verwenden, wenn ein Trigger die benutzerdefinierte Antwort nach erfolgter Ausführung senden soll.

Einige Beispiele für Inhalte, die Sie in Form von benutzerdefinierten Antworten senden können: Benutzerdefinierte Fehlermeldungen, eine Etikettenvorschau, erzeugte PDF-Dateien, Druckstrom-Dateien

(Spool-Dateien), XML-Dateien mit Daten aus der Druck-Engine und der Etikettenvorschau (als Base64-Zeichenfolge codiert) usw. Die Möglichkeiten sind praktisch unbegrenzt.

Der Hauptteil der Definition lautet:

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerWithResponse_InputMessage">
    <wsdl:part name="text" type="xsd:string"/>
    <wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="WebSrviTrg_ExecuteTriggerWithResponse_OutputMessage">
    <wsdl:part name="ExecuteTriggerWithResponseResult" type="xsd:int"/>
    <wsdl:part name="responseData" type="xsd:base64Binary"/>
    <wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Im obigen Beispiel gibt es zwei Eingabevariablen (Sie geben ihre Werte an):

- **text:** Filter in der Konfiguration parst diese Eingabezeichenfolge. Normalerweise ist die Eingabezeichenfolge als CSV- oder XML-Datei strukturiert, wodurch sie sich einfach parsen lässt. Sie können auch ein anderes Textdateiformat verwenden.
- **wait:** Dies ist ein boolesches Feld, das zwei Dinge festlegt:
  - Ob Sie auf die Statusantwort des Druckauftrags warten möchten oder nicht.
  - Ob der Webdienst ein Feedback ausgeben soll oder nicht.

  Wählen Sie für *Wahr* den Wert **1**. Wählen Sie für *Falsch* den Wert **0**. Abhängig vom ausgewählten Methodentyp gibt es entweder eine vordefinierte Antwort, oder Sie können eine benutzerdefinierte Antwort senden.

Außerdem sind im Beispiel die folgenden optionalen Ausgabevariablen enthalten.

ANMERKUNG

Sie erhalten Werte von optionalen Ausgabevariablen, wenn Sie sie durch Einstellen des **wait**-Feldwerts auf **1** anfordern.

- **ExecuteTriggerWithResponseResult:** Die Ganzzahl-Antwort enthält den Wert 0, wenn bei der Datenverarbeitung keine Fehler aufgetreten sind. Sie enthält eine Ganzzahl größer 0, wenn Fehler aufgetreten sind. Die Anwendung, die den Webdienst-Aufruf an NiceLabel Automation ausführt, kann diese Antwort als Fehlerindikator verwenden.
- **responseData:** Benutzerdefinierte Antwort, die Sie bei der Konfiguration des Webdienst-Triggers festlegen können. Die Antwort erfolgt in Form von base64-codierten Daten.
- **errorText:** Wenn ein Fehler bei der Trigger-Verarbeitung auftritt, enthält diese Zeichenfolge den Antwort-Wert für den Druckauftrags-Status.

> **ANMERKUNG**
> Fall bei der Trigger-Verarbeitung ein Fehler gemeldet wird, beinhaltet die XML-Antwortnachricht das **errorText**-Element. Der Wert dieses Elements enthält die Fehlerbeschreibung. Ist jedoch kein Fehler aufgetreten, wird das Element nicht in die XML-Antwort eingeschlossen.

ExecuteTriggerAndSetVariables-Methode

Der Hauptteil der Definition lautet:

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariables_InputMessage">
    <wsdl:part name="text" type="xsd:string"/>
    <wsdl:part name="variableData" type="xsd:string"/>
    <wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
    <wsdl:message
name="WebSrviTrg_ExecuteTriggerAndSetVariables_OutputMessage">
    <wsdl:part name="ExecuteTriggerAndSetVariablesResult" type="xsd:int"/>
    <wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Im obigen Beispiel gibt es drei Eingabevariablen (Sie geben ihre Werte an):

- **text:** Filter in der Konfiguration parst diese Eingabezeichenfolge. Normalerweise ist die Eingabezeichenfolge als CSV- oder XML-Datei strukturiert, wodurch sie sich einfach parsen lässt. Sie können auch ein anderes Textdateiformat verwenden.
- **wait:** Dies ist ein boolesches Feld, das zwei Dinge festlegt:
  - Ob Sie auf die Statusantwort des Druckauftrags warten möchten oder nicht.
  - Ob der Webdienst ein Feedback ausgeben soll oder nicht.

  Wählen Sie für *Wahr* den Wert `1`. Wählen Sie für *Falsch* den Wert `0`. Abhängig vom ausgewählten Methodentyp gibt es entweder eine vordefinierte Antwort, oder Sie können eine benutzerdefinierte Antwort senden.

- **variableData:** Dies ist die Zeichenfolge, welche die *Name:Wert*-Paare enthält. Der Trigger liest alle Paare aus und ordnet die verfügbaren Werte den gleichnamigen Trigger-Variablen zu. Ist die Variable im Trigger nicht vorhanden, verwirft der Trigger das jeweilige *Name:Wert*-Paar. Wenn Sie die Liste von Variablen und ihren Werten mithilfe dieser Methode angeben, müssen Sie keine Datenextraktion in den Filtern definieren. Der Trigger übernimmt das gesamte Parsing für Sie.
  Es gibt zwei verfügbare Strukturen für den variableData-Inhalt.

  XML-Struktur

  Der Trigger stellt Variablen innerhalb des `<Variables />`-Stammelements der XML-Datei bereit. Der Variablenname enthält den Attributnamen und der Variablenwert enthält den Elementwert.

  ```
  <?xml version="1.0" encoding="utf-8"?>
  <Variables>
  ```

```
    <variable name="Variable1">Value 1</variable>
    <variable name="Variable2">Value 2</variable>
    <variable name="Variable3">Value 3</variable>
</Variables>
```

ANMERKUNG

Betten Sie Ihre XML-Daten innerhalb des CDATA-Abschnitts ein. **CDATA**, also **Zeichendaten**, ist ein Abschnitt des Elementinhalts, der dem Parser mitteilt, dass er als normale Zeichendaten interpretiert werden soll, nicht als Markup. Daher behandelt der Trigger den gesamten Inhalt als Zeichendaten. **`<element>ABC</element>`** wird z. B. als **`&lt;element&gt;ABC&lt;/element&gt;`** interpretiert. Jeder CDATA-Abschnitt beginnt mit der Folge **`<![CDATA[`** und endet mit der Folge **`]]>`**. Fügen Sie Ihre XML-Daten einfach zwischen diesen Folgen ein.

Name-Wert-Paare

Der Trigger stellt Variablen in Form eines Textdatenstroms bereit. Jedes *Name:Wert*-Paar steht in einer eigenen Zeile. Der Variablenname steht auf der linken Seite des Gleichheitszeichens (=), der Variablenwert auf der rechten Seite.

```
Variable1="Value 1"
Variable2="Value 2"
Variable3="Value 3"
```

Dies sind die optionalen Ausgabevariablen:

ANMERKUNG

Sie erhalten Werte für optionale Variablen, wenn Sie sie durch Einstellen von **wait** auf **`1`** anfordern:

- **ExecuteTriggerAndSetVariablesResult:** Die Ganzzahl-Antwort enthält den Wert 0, wenn bei der Datenverarbeitung keine Fehler aufgetreten sind. Sie enthält eine Ganzzahl größer 0, wenn bei der Datenverarbeitung Fehler gemeldet werden. Die Anwendung, die den Webdienst-Aufruf an NiceLabel Automation ausführt, kann die Antwort als Fehlerindikator verwenden.
- **errorText:** Dieser Zeichenfolgenwert enthält eine Druckauftrags-Statusantwort, wenn ein Fehler während der Trigger-Verarbeitung auftritt.

ANMERKUNG

Im Fall eines Trigger-Verarbeitungsfehlers ist dieses Element in der XML-Antwortnachricht enthalten. Sein Wert beinhaltet die Fehlerbeschreibung. Ist jedoch kein Fehler aufgetreten, wird das Element nicht in die XML-Antwort eingeschlossen.

ExecuteTriggerAndSetVariablesWithResponse-Methode

Sie würden diese Methode verwenden, wenn der Trigger nach erfolgter Ausführung eine benutzerdefinierte Antwort senden soll.

Einige Beispiele für Inhalte, die Sie in Form von benutzerdefinierten Antworten senden können: Benutzerdefinierte Fehlermeldungen, eine Etikettenvorschau, erzeugte PDF-Dateien, Druckstrom-Dateien (Spool-Dateien), XML-Dateien mit Daten aus der Druck-Engine und der Etikettenvorschau (als Base64-Zeichenfolge codiert) usw. Die Möglichkeiten sind praktisch unbegrenzt.

Der Hauptteil der Definition lautet:

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariablesWithResponse_
InputMessage">
    <wsdl:part name="text" type="xsd:string"/>
    <wsdl:part name="variableData" type="xsd:string"/>
    <wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariablesWithResponse_
OutputMessage">
    <wsdl:part name="ExecuteTriggerAndSetVariablesWithResponseResult"
type="xsd:int"/>
    <wsdl:part name="responseData" type="xsd:base64Binary"/>
    <wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Es gibt drei Eingabevariablen (Sie geben ihre Werte an):

- **text:** Filter in der Konfiguration parst diese Eingabezeichenfolge. Normalerweise ist die Eingabezeichenfolge als CSV- oder XML-Datei strukturiert, wodurch sie sich einfach parsen lässt. Sie können auch ein anderes Textdateiformat verwenden.
- **wait:** Dies ist ein boolesches Feld, das zwei Dinge festlegt:
  - Ob Sie auf die Statusantwort des Druckauftrags warten möchten oder nicht.
  - Ob der Webdienst ein Feedback ausgeben soll oder nicht.

  Wählen Sie für *Wahr* den Wert **`1`**. Wählen Sie für *Falsch* den Wert **`0`**. Abhängig vom ausgewählten Methodentyp gibt es entweder eine vordefinierte Antwort, oder Sie können eine benutzerdefinierte Antwort senden.

- **variableData:** Dies ist die Zeichenfolge, welche die *Name:Wert*-Paare enthält. Der Trigger liest alle Paare aus und ordnet die verfügbaren Werte den gleichnamigen Trigger-Variablen zu. Ist die Variable im Trigger nicht vorhanden, verwirft der Trigger das jeweilige *Name:Wert*-Paar. Wenn Sie die Liste von Variablen und ihren Werten mithilfe dieser Methode angeben, müssen Sie keine Datenextraktion in den Filtern definieren. Der Trigger übernimmt das gesamte Parsing für Sie.
  Es gibt zwei verfügbare Strukturen für den variableData-Inhalt.

  XML-Struktur

  Der Trigger stellt Variablen innerhalb des **`<Variables />`**-Stammelements der XML-Datei bereit. Der Variablenname enthält den Attributnamen und der Variablenwert enthält den Elementwert.

```
<?xml version="1.0" encoding="utf-8"?>
<Variables>
    <variable name="Variable1">Value 1</variable>
    <variable name="Variable2">Value 2</variable>
    <variable name="Variable3">Value 3</variable>
</Variables>
```

ANMERKUNG

Betten Sie Ihre XML-Daten innerhalb des CDATA-Abschnitts ein. **CDATA**, also **Zeichendaten**, ist ein Abschnitt des Elementinhalts, der dem Parser mitteilt, dass er als normale Zeichendaten interpretiert werden soll, nicht als Markup. Daher behandelt der Trigger den gesamten Inhalt als Zeichendaten. **`<element>ABC</element>`** wird z. B. als **`&lt;element&gt;ABC&lt;/element&gt;`** interpretiert. Jeder CDATA-Abschnitt beginnt mit der Folge **`<![CDATA[`** und endet mit der Folge **`]]>`**. Fügen Sie Ihre XML-Daten einfach zwischen diesen Folgen ein.

Name-Wert-Paare

Der Trigger stellt Variablen in Form eines Textdatenstroms bereit. Jedes *Name:Wert*-Paar steht in einer eigenen Zeile. Der Variablenname steht auf der linken Seite des Gleichheitszeichens (=), der Variablenwert auf der rechten Seite.

```
Variable1="Value 1"
Variable2="Value 2"
Variable3="Value 3"
```

Dies sind die optionalen Ausgabevariablen:

ANMERKUNG

Sie erhalten ihre Werte, wenn Sie sie durch Einstellen von **wait** auf **`1`** anfordern:

- **ExecuteTriggerAndSetVariablesWithResponseResult:** Die Ganzzahl-Antwort enthält den Wert 0, wenn bei der Datenverarbeitung keine Fehler aufgetreten sind. Sie enthält eine Ganzzahl größer 0, wenn bin der Antwort Fehler gemeldet werden. Die Anwendung, welche den Webdienst-Aufruf an NiceLabel Automation ausführt, kann die Antwort als Fehlerindikator verwenden.
- **responseData:** Benutzerdefinierte Antwort, die Sie bei der Konfiguration des Webdienst-Triggers festlegen können. Die Antwort erfolgt in Form von base64-codierten Daten.
- **errorText:** Dieser Zeichenfolgenwert enthält eine Druckauftrags-Statusantwort, wenn ein Fehler während der Trigger-Verarbeitung auftritt.

> **ANMERKUNG**
> Im Fall eines Trigger-Verarbeitungsfehlers ist dieses Element in der XML-Antwortnachricht enthalten. Sein Wert beinhaltet die Fehlerbeschreibung. Ist jedoch kein Fehler aufgetreten, wird das Element nicht in die XML-Antwort eingeschlossen.

Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikationsparameter fest, durch die Sie Feedback von der Druck-Engine erhalten können.

> **PRODUKTEBENEN-INFO**
> Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

- **Überwachtes Drucken:** Aktiviert den synchronen Druckmodus. Verwenden Sie ihn, wenn Sie den Status des Druckauftrags an die Drittanwendung zurücksenden wollen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Die Optionen im Abschnitt **Datenverarbeitung** geben an, ob Sie die Daten kürzen möchten, damit sie in eine Variable passen, oder ob die fehlenden Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet Fehler und unterbricht den Druckprozess, wenn Sie versuchen, zu lange Werte in Etikettenvariablen zu speichern oder Werte für nicht vorhandene Etikettenvariablen anzugeben.

- **Übermäßig lange Variableninhalte ignorieren:** kürzt Datenwerte, die die Länge der Variable gemäß Definition im Etiketten-Designer überschreiten, um sie auf passende Länge zu bringen. Diese Option wird wirksam, wenn Sie Variablenwerte in Filtern oder aus Befehlsdateien festlegen und wenn Sie Werte von Triggervariablen den gleichnamigen Etikettenvariablen zuordnen.

> **Beispiel**
>
> Die Etikettenvariable akzeptiert maximal 5 Zeichen. Wenn diese Option aktiviert ist, wird jeder Wert über 5 Zeichen auf die ersten 5 Zeichen gekürzt. Wenn der Wert 1234567 lautet, ignoriert die 6. Und 7. Stelle.

- **Fehlende Etikettenvariablen ignorieren:** Beim Drucken mit Befehlsdateien (z. B. JOB-Dateien), werden alle Variablen ignoriert, die:
  - in der Befehlsdatei angegeben sind (anhand des SET Befehls)
  - nicht auf dem Etikett definiert sind

  Etwas Ähnliches geschieht, wenn Sie einen Zuweisungsbereich in einem Filter definieren, um alle Name-Wert-Paare zu extrahieren, Ihr Etikett jedoch weniger Variablen enthält.
  Wenn Sie Werte von nicht vorhandenen Etikettenvariablen einstellen, gibt einen Fehler aus. Wenn diese Option aktiviert ist, wird der Druck fortgesetzt.

Die Optionen im Abschnitt **Scripting** geben die Scripting-Möglichkeiten an.

- **Scripting-Sprache:** Wählt die Scripting-Sprache für den Trigger aus. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden die ausgewählte Scripting-Sprache.

Die Optionen im Abschnitt **Empfangene Daten speichern** legen die Befehle fest, die für vom Trigger empfangene Daten zur Verfügung stehen.

- **Vom Trigger empfangene Daten speichern in Datei:** Aktivieren Sie diese Option, um vom Trigger empfangene Daten zu speichern. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und den Dateinamen enthält.
- **Bei Fehler vom Trigger empfangene Daten speichern in Datei:** Aktivieren sie diese Option, um die vom Trigger empfangenen Daten nur dann zu speichern, wenn ein Fehler bei der Ausführung der Aktion auftritt. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu behalten, die das Problem hervorgerufen haben, um die Lösungsfindung zu erleichtern.

ANMERKUNG

Stellen Sie sicher, dass Sie die Unterstützung für Überwachtes Drucken aktivieren. Ist sie nicht aktiviert, kann Fehler bei der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

ANMERKUNG

speichert die empfangenen Daten in einer temporären Datei. Diese temporäre Datei wird sofort nach Abschluss der Trigger-Ausführung gelöscht. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

Sicherheit

- **Trigger sperren und verschlüsseln:** Aktiviert den Trigger-Schutz. Wenn Sie die Option aktivieren, wird der Trigger gesperrt und kann nicht mehr bearbeitet werden. Dadurch werden die Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

3.8. Cloud-Trigger

PRODUKTEBENEN-INFO

Ein Abonnement von **Label Cloud** ist erforderlich.

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

Hier erfahren Sie mehr über NiceLabel Label Cloud.

Cloud-Trigger ermöglicht Ihnen die Integration von NiceLabel Cloud oder von Ihrem vor Ort gehosteten Control Center mit vorhandenen Geschäftssystemen, die in privaten Clouds oder in speziellen

Rechenzentren ausgeführt werden. Wenn ein vorhandenes Geschäftssystem (wie SAP S/4HANA oder Oracle NetSuite) eine Ausgabe erzeugt, ermöglicht Ihnen eine in der Cloud gehostete API, HTTP-Anforderungen an den Cloud-Trigger zu senden.

Der Cloud-Trigger ermöglicht Ihnen das lokale Drucken von Etiketten, deren Inhalte aus den Cloud-basierten Informationssystemen stammen. Da der im lokalen Automation ausgeführte Cloud-Trigger Standardmethoden für den Zugriff auf Cloud-basierte Dienste verwendet, können Sie lokales Drucken auf sichere und zeitsparende Weise bereitstellen.

Der Cloud-Trigger bietet eine sichere und transparente Möglichkeit zur Integration Ihres lokalen Etikettendrucks anhand von Anwendungen, die über das offene Internet kommunizieren.

Im Gegensatz zum HTTP Server Trigger müssen Sie für den Cloud-Trigger keine eingehenden Ports in Ihrer Firewall öffnen. Der Cloud-Trigger nutzt eine dedizierte NiceLabel API, die in der Cloud ausgeführt wird. Deswegen erfordert der Trigger nur eine Öffnung des ausgehenden Ports 443 oder der Ports 9350-9354. In den meisten Fällen sind diese Ports bereits geöffnet.

Beim Implementieren des Cloud-Triggers haben Sie zwei Optionen:

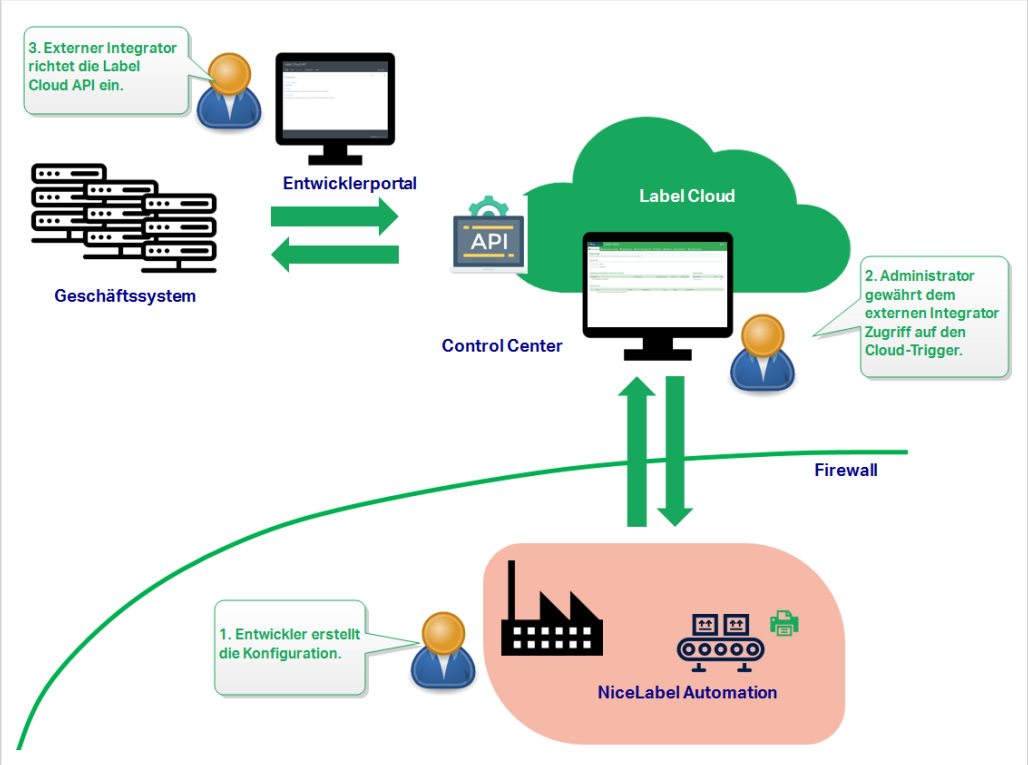
- Sie können den Trigger in Ihrer implementieren NiceLabel Cloud.
- Sie können den Trigger in Ihrem vor Ort gehosteten Control Center implementieren, das entweder lokal auf Ihren Servern oder in einer privaten Cloud-Infrastruktur ausgeführt wird.

In Bezug auf den Funktionsumfang sind beide Optionen gleichwertig. Wählen Sie Ihre bevorzugte Option auf Basis Ihrer verfügbaren Infrastruktur.

3.8.1. Cloud-Trigger mit NiceLabel Cloud implementieren

3.8.1.1. Implementierungsphasen für NiceLabel Cloud

Um lokalen Etikettendruck anhand des in NiceLabel Cloud implementierten Cloud-Triggers zu ermöglichen, müssen Sie eine Zusammenarbeit zwischen Benutzern mit drei Rollen herstellen: ein Benutzer, der den Cloud-Trigger auf dem lokalen Automation Server konfiguriert (Entwickler), ein Benutzer, der den Cloud-Trigger in NiceLabel NiceLabel Cloud einrichtet und ein Benutzer, der das Abonnement im Entwicklerportal vornimmt.



1. Der **Entwickler** übernimmt die Konfiguration und Bereitstellung der Cloud-Trigger-Konfiguration auf dem lokalen Automation Server anhand von Automation Builder und Automation Manager.

   > **ANMERKUNG**
   > NiceLabelAutomation muss in der NiceLabel Cloud angemeldet sein.

   Im Abschnitt Cloud-Trigger konfigurieren in Automation Builder finden Sie weitere Informationen.

2. Der **NiceLabel Cloud Administrator** gewährt dem externen Integrator Zugriff auf den Cloud-Trigger in Control Center. Nach Fertigstellung sendet der NiceLabel Cloud Administrator dem externen Integrator den entsprechenden Integratorschlüssel.
   Siehe Abschnitt Cloud-Trigger-Zugriff für den externen Integrator einrichten für mehr Details.

3. Der **externe Integrator** führt im Entwicklerportal das Geschäftssystem des Kunden und die NiceLabel Cloud zusammen.

   > **ANMERKUNG**
   > Im Entwicklerportal wird die dafür erstellte API namens **Cloud Trigger** gehostet. Diese API fungiert als Schnittstelle zwischen den Ereignissen, die im Geschäftssystem des Kunden stattfinden, und der lokale ausgeführten Automation Konfiguration.

> **ANMERKUNG**
> Der Begriff „extern“ bedeutet, dass die Rolle dieses Benutzers darin besteht, das Abonnement im Entwicklerportal vorzunehmen. Das erstellte Abonnement authentifiziert den Integrator. Externe Integratoren müssen nicht zwangsläufig Personen außerhalb des Unternehmens sein. Die Aufgaben können auch von internen Integratoren übernommen werden, die dem Entwicklungsteam des Unternehmens angehören.

Der externe Integratoren führt folgende Aktionen im Entwicklerportal aus:

a. Anmeldung im Entwicklerportal. Vor der ersten Anmeldung muss der Integrator außerdem den Registrierungsvorgang durchlaufen.

b. Erstellen eines Abonnements für die Cloud Trigger API.

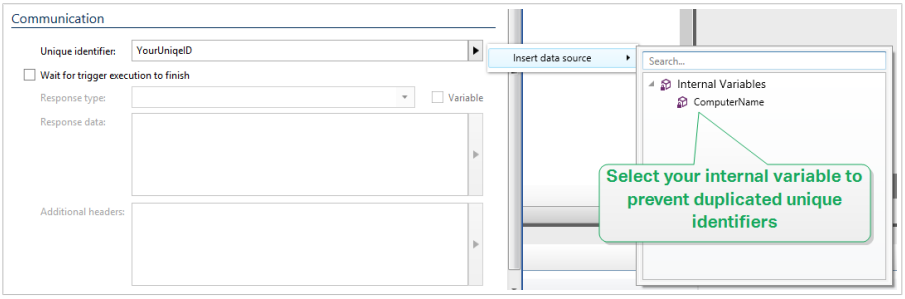
c. Verknüpfung des Abonnements mit dem Integratorschlüssel. So erhält das Abonnement Zugriff auf die Cloud-Trigger des Kunden.
Weitere Details finden Sie im Abschnitt NiceLabel Cloud API im Entwicklerportal einrichten.

3.8.1.2. Cloud-Trigger in Automation Builder konfigurieren

Dieser Abschnitt beschreibt, wie Sie den Cloud-Trigger in Automation konfigurieren, das auf Ihrem lokalen Server ausgeführt wird.

1. Öffnen Sie Ihren Automation Builder. Stellen Sie sicher, dass der Automation Builder bei der NiceLabel Cloud angemeldet ist. Gehen Sie auf **Datei** > **Über** > **NiceLabel Cloud**, um sich zu vergewissern, dass Sie angemeldet sind.
2. Die Registerkarte **Konfigurationselemente** wird geöffnet. Klicken Sie auf **Cloud-Trigger**, um eine neue Konfiguration für den Cloud-Trigger zu erstellen.
3. Stellen Sie **Namen** und **Beschreibung** ein, um Ihren Cloud-Trigger unter anderen Triggern leicht finden zu können.
4. Nehmen Sie die **Kommunikation**-Einstellungen für den Trigger vor:
   - Legen Sie die **Eindeutige Kennung** fest. Nachdem Sie den Trigger bereitgestellt haben, ist diese eindeutige Kennung erforderlich, um den Trigger aufzurufen.
     Wenn Sie die Cloud-Trigger-Konfiguration auf mehreren Computern durchführen, müssen Sie sicherstellen, dass jeder Computer automatisch seine eigene eindeutige Kennung verwendet. Um unerwünschte Duplikate zu vermeiden, fügen Sie interne Variablen als Teil der **Eindeutigen Kennung** ein. Sie können zu diesem Zweck zwei interne Variablen verwenden:
     - **ComputerName**: Der Name des Computers, auf dem die Konfiguration ausgeführt wird.
     - **SystemUserName**: Der Windows-Benutzername des aktuell angemeldeten Benutzers.

     Um interne Variablen in die **Eindeutige Kennung** einzufügen, klicken Sie auf „Datenquelle einfügen“ und wählen Sie Ihre internen Variablen aus.



- **Warten, bis Trigger-Ausführung abgeschlossen ist:** Das HTTP-Protokoll wartet darauf, dass der Empfänger (in diesem Fall NiceLabel Automation) eine numerische Antwort an den Absender übermittelt, die den Status der empfangenen Nachricht anzeigt. Standardmäßig antwortet NiceLabel Automation mit dem Code 200. Dies zeigt an, dass Automation die Daten erfolgreich empfangen hat, gibt aber keine Informationen über den Erfolg von Trigger-Aktionen.

  Diese Option legt fest, dass ein Trigger nicht sofort nach Empfang der Daten eine Antwort sendet, sondern wartet, bis alle Aktionen ausgeführt wurden. Danach sendet er den Antwort-Code, um das erfolgreiche Ausführen von Aktionen zu bestätigen. Wenn diese Option aktiviert ist, können Sie eine benutzerdefinierte Antwort und die entsprechenden Daten zurücksenden (z. B. ist die Antwort auf eine HTTP-Abfrage die Etikettenvorschau im PDF-Format).

  Beim Cloud-Trigger sind die relevanten, in Automation integrierten Standard-HTTP-Antwortcodes:

HTTP-Antwortcode	Beschreibung
200	Alle Aktionen erfolgreich ausgeführt.
500	Beim Ausführen der Aktion sind Fehler aufgetreten.

> **ANMERKUNG**
>
> Um Feedback zum Druckprozess an Automation zu senden, aktivieren Sie den **synchronen** Druckmodus. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

- **Antworttyp:** Gibt den Typ Ihrer Antwortnachricht an. Häufig verwendete Internet-Medientypen (auch MIME-Typen oder Content-Typen genannt) stehen in der Dropdown-Liste zur Verfügung. Falls Ihr Medientyp nicht in der Liste enthalten ist, geben Sie ihn selbst ein. Automation sendet die ausgehenden Antwortdaten im Format des festgelegten Medientyps als Feedback. **Variable** aktiviert variable Medientypen. Wählen Sie bei aktivierter Option eine Variable aus (oder erstellen Sie eine Variable), die den Medientyp enthält.

> **ANMERKUNG**
>
> Wenn Sie keinen MIME-Typ angeben, verwendet NiceLabel Automation `application/octet-stream` als Standard.

- **Antwortdaten:** Definiert den Inhalt Ihrer Antwortnachricht. Beispiele für Inhalte, die Sie in Form von HTTP-Antworten senden können: Benutzerdefinierte Fehlermeldungen, Etikettenvorschau, erzeugte PDF-Dateien, Druckstrom-(Spool-)Dateien, XML-Dateien mit Daten aus der Druck-Engine und der Etikettenvorschau (als Base64-Zeichenfolge codiert) usw.
  Wenn Ihre Ausgabe aus binären Inhalten besteht (z. B. Etikettenvorschau oder Druckstrom), müssen Sie einen unterstützten Medientyp auswählen, z. B. **`image/jpeg`** oder **`application/octet-stream`**.

- **Zusätzliche Header**: Ermöglicht es Ihnen, benutzerdefinierte MIME-Header für die HTTP-Antwortnachricht zu definieren.
  Die Syntax für Antwortheader und Beispiele stehen im Abschnitt Aktion „HTTP-Anfrage“ zur Verfügung.

  TIPP

  Sie können bei **Antwortdaten** und **zusätzlichen Headern** festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzufügen, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie Ihre Variable aus der Liste ein. Sie können auch eine neue Variable erstellen, die die Daten enthält, sie Sie verwenden möchten. Weitere Informationen finden Sie im Abschnitt Zusammengesetzte Werte verwenden im Benutzerhandbuch.

5. Stellen Sie den Trigger in Automation Manager bereit und starten Sie ihn. Der Cloud-Trigger überwacht jetzt eingehende Anfragen.

ANMERKUNG

Wenn Ihre Konfiguration höhere Verfügbarkeit und Skalierbarkeit erfordert, können Sie mehrere identische Cloud-Trigger bereitstellen. Installieren Sie zu diesem Zweck mehrere Instanzen von Automation und stellen Sie die Cloud-Trigger auf ihnen bereit. Wenn die bereitgestellten Cloud-Trigger dieselbe **Eindeutige Kennung** haben, verteilt der integrierte Lastausgleich in NiceLabel Cloud die Traffic-Last gleichmäßig auf sie.

3.8.1.3. Cloud-Trigger-Zugriff für den externen Integrator einrichten

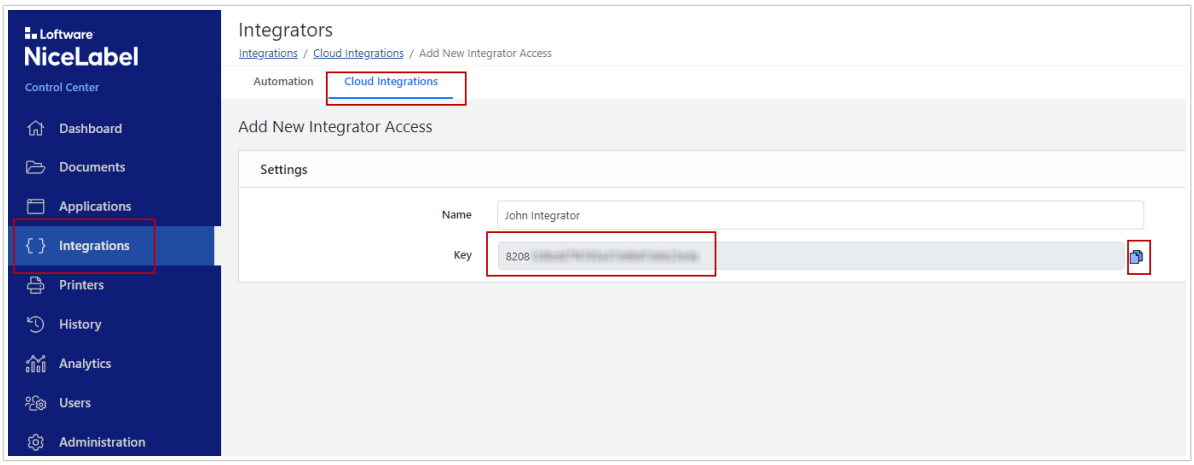
ANMERKUNG

Um den Zugriff des Integrators auf den Cloud-Trigger einzurichten, müssen Sie die **Berechtigung zur Verwaltung der Cloud-Integration** im Cloud-basiertenControl Center haben. Im Benutzerhandbuch für Control Center finden Sie weitere Informationen zur Verwaltung von Benutzerrechten.

1. Gehen Sie zu Ihrem Cloud-basierten Control Center. Öffnen Sie den Webbrowser und geben Sie https://<yourlabelcloudname>./dashboard ein:

   ```
   https://<yourlabelcloudname>.onnicelabel.com/dashboard.
   ```

2. Gehen Sie auf **Anwendungen** > **Cloud-Integrationen**.
3. Klicken Sie auf **+Hinzufügen**. Die Seite **Cloud-Trigger-Integratoren** wird geöffnet.
4. Geben Sie den **Namen** des Integrators ein, den Sie hinzufügen.



5. Kopieren Sie den **Schlüssel**.
6. Klicken Sie auf **Speichern**.
7. Verweisen Sie den externen Integrator auf das Entwicklerportal. Senden Sie die folgenden Informationen an den externen Integrator:

   - Link zur NiceLabel Cloud API: https://developerportal.onnicelabel.com/
   - Den Integrator-Schlüssel (siehe Schritt 5).
   - Die eindeutige Kennung des Triggers. Sie finden diese **Eindeutige Kennung** in den Automation-Konfigurationseinstellungen für Cloud-Trigger (siehe Schritt 4 im Abschnitt Cloud-Trigger in konfigurieren Automation Builder.

> **ANMERKUNG**
> Externe Integratoren benötigen den Schlüssel, um sich zum Aufrufen des Cloud-Triggers des Kunden zu authentifizieren.

> **ANMERKUNG**
> Weitere Informationen finden Sie im Abschnitt „Cloud-Trigger“ Ihres Control Center Benutzerhandbuchs.

3.8.1.4. Ein Abonnement im Entwicklerportal erstellen

Nach Erhalt der erforderlichen Informationen vom NiceLabel Cloud-Administrator muss sich der externe Integrator zuerst im Entwicklerportal registrieren und Abonnements (für jeden Kunden) zum Aufrufen der Trigger erstellen. Diese Triggeraufrufe stammen aus den Cloud-basierten Informationssystemen der Kunden.

ANMERKUNG

Wenn Sie die E-Mail des Entwicklerportals nicht in Ihrem Posteingang finden, sehen Sie im Spam-Ordner nach.

ANMERKUNG

Kunden sind Unternehmen, die Cloud-Trigger-Konfigurationen ausführen, welche die Daten von externen Informationssystemen erhalten.

ANMERKUNG

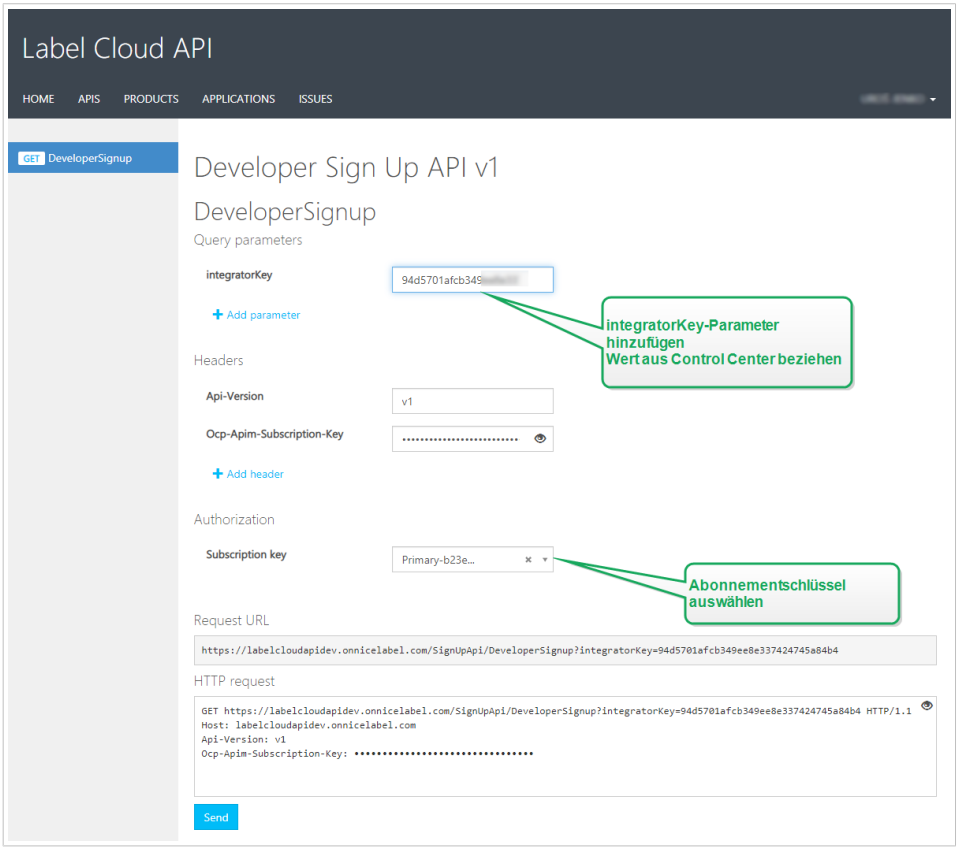
Jeder Integrator kann mit einem Abonnement mehrere Cloud-Trigger aufrufen.

1. Öffnen Sie Ihren Browser und gehen Sie auf https://developerportal.onnicelabel.com/
2. Folgen Sie den Anweisungen auf dem Bildschirm, um die Registrierung durchzuführen. Nach Klicken auf **Sign up** erhalten Sie eine Bestätigungs-E-Mail. Klicken Sie auf den Bestätigungslink, um Ihren Account im Entwicklerportal zu aktivieren.
3. Öffnen Sie die Registerkarte **Produkte** und klicken Sie auf **Label Cloud**. Danach werden Sie auf eine Seite mit Ihren APIs und vorhandenen Abonnements geleitet.
4. Klicken Sie auf **Add subscription**. Die Seite **Subscribe to product** wird geöffnet.

   ANMERKUNG

   Sie können mehrere Abonnements erstellen. Ein einzelnes Abonnement kann jedoch nur für einen einzelnen Kunden verwendet werden. Daher empfiehlt NiceLabel Ihnen, den Namen den Kunden in den **Subscription name (Abonnementnamen)** einzubeziehen, z. B. `Cloud Trigger Example Customer`.
5. Geben Sie den **Subscription name** ein.
6. Klicken Sie auf **Confirm**. Das neu erstellte Abonnement steht unter **Products** > **Label Cloud verfügbar**.
7. Klicken Sie auf der Seite **Label Cloud** auf **Developer Sign Up API v1**.
8. Klicken Sie auf **Try it**. Die API-Seite wird geöffnet.



9. Fügen Sie den **Integrator Key** aus dem Cloud-basierten Control Center des Kunden ein.
10. Klicken Sie auf **Send**.
    - Die Antwort lautet: `Abonnement <Ihr Abonnementschlüssel> erfolgreich mit Integratorschlüssel <Wert des Integratorschlüssels> verbunden.`

> **ANMERKUNG**
> Sie haben den Integratorschlüssel vom NiceLabel Cloud Administrator erhalten. Der Schlüssel sieht folgendermaßen aus:
> `979d7be5df2b473193ac5519f94cd901`

Beispiel

Wenn Sie den Integrationsschlüssel als Abfrageparameter weitergeben, sieht die URL folgendermaßen aus:`https://labelcloudapi.onnicelabel.com/SignUpApi/DeveloperSignup?integratorKey=979d7be5df2b473193ac5519f94cd901.`

Wenn Sie den Aufruf anhand der URL (wie im Beispiel gezeigt) vornahmen, gleicht die **DeveloperSignup**-Operation das Abonnement mit dem zugeordneten Kunden ab. Auf diese Weise authentifiziert sich der Integrator beim Aufrufen des Cloud-Triggers, der in der Automation des Kunden ausgeführt wird.
Die verbundenen Abonnements werden außerdem im Cloud-basierten Control Center angezeigt. Prüfen Sie unter **Anwendungen** > **Cloud-Integrationen**, ob der externe Integrator sein Abonnement

verbunden hat. Die Cloud-Integration sollte den Status **Entwickler [Name, E-Mail-Adresse] abonniert** haben.

3.8.1.5. Aufrufen Ihres Cloud-Triggers (NiceLabel Cloud Implementierung)

Mit diesem Schritt stellen Sie sicher, dass die Ausgaben der externen Geschäftssysteme lokal gehostete Cloud-Trigger erfolgreich ausführen. Dies ist der Zweck der **CloudTrigger**Operation. Geben Sie in der URL für den Aufruf den Namen des Triggers an, den Sie aufrufen.

Um einen Trigger mit der eindeutigen Kennung `MyCloudTrigger` aufzurufen, verwenden Sie die folgende URL:

`https://labelcloudapi.onnicelabel.com/TriggerApi/CloudTrigger/MyCloudTrigger`

Rufen Sie die URL für jedes Ereignis (Ausgabe) im externen Geschäftssystem wie im Beispiel gezeigt auf. Jeder Aufruf führt den Cloud-Trigger aus, der auf dem lokalen Automation Server gehostet wird.

Alle API-Aufrufe müssen die beiden folgenden Header enthalten:

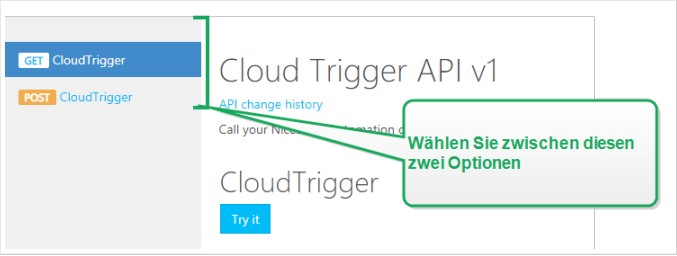
- **Api-Version** ist die Version der API, die Sie verwenden. Aktuell ist v1 die einzig verfügbare API-Version.
- **Ocp-Apim-Subscription-Key** ist der Schlüssel, der Ihr Abonnement identifiziert.

CloudTrigger-Aufrufe testen

Um sich mit der Funktion von **CloudTrigger**-Aufrufen vertraut zu machen, können Sie solche Aufrufe im Entwicklerportal testen.

Bevor Sie diesen Aufruf testen, müssen Sie eine funktionierende Automation Konfiguration einrichten.

1. Öffnen Sie das **Entwicklerportal**, öffnen Sie die **Products**-Registerkarte und klicken Sie auf **Label Cloud**.
2. Wählen Sie **Cloud Trigger API v1**.
3. Erstellen Sie ein Beispiel für GET- oder POST-Methoden. Klicken Sie auf den entsprechenden Link.
    - Klicken Sie nach Auswahl der Methode auf **Try it**. Eine neue Seite wird geöffnet. Die **triggerID** wird bereits unter den **Query parameters** aufgeführt.
    - Geben Sie im **Value**-Feld per Kopieren und Einfügen die **triggerID** ein, die Sie vom Entwickler der Automation-Konfiguration erhalten haben. Dies ist die **Eindeutige Kennung** des Triggers. Die **Eindeutige Kennung** steht unter **Automation Builder** > **Triggereinstellungen** > **Allgemein** zur Verfügung.



4. Wählen Sie unter **Authorization** den Abonnementschlüssel aus. Da Sie bereits mindestens ein Abonnement definiert haben, enthält die Dropdown-Liste bereits den Schlüssel für das definierte Abonnement. Wählen Sie diesen (primären oder sekundären) Schlüssel aus.
5. Klicken Sie auf **Send**.
    - Der **Antwortstatus (Response status)** lautet „200 OK“.

3.8.1.6. Schnelle Überprüfung Ihres Cloud-Triggers

Wenn Sie Ihre NiceLabel Cloud-API im Entwicklerportal eingerichtet haben, können Sie eine einfache Konfiguration in Automation Builder vornehmen, um zu testen, ob der Cloud-Trigger funktioniert. Ist dies der Fall, erhalten Sie die Meldung „Trigger works ok.“ auf der Seite „Cloud Trigger API“, nachdem Sie auf **Try it** geklickt haben.

1. Öffnen Sie den Automation Builder erstellen Sie eine neue Konfiguration. Stellen Sie sicher, dass Ihr Automation Manager mit der NiceLabel Cloud verbunden ist.
2. Fügen Sie einen neuen **Cloud-Trigger** hinzu.
3. Legen Sie den **Namen** und die **Beschreibung** sowie die **Eindeutige Kennung** fest. In diesem Fall verwenden wir `TestCloudTrigger` als eindeutige Kennung.
4. Aktivieren Sie **Warten, bis Trigger-Ausführung abgeschlossen ist**. So können Sie Trigger-Antworten zurückverfolgen.
    - Wählen Sie **text/plain** als **Antworttyp**.
    - Definieren Sie die **Antwortdaten**. Dies sind die Daten, die Sie empfangen werden, wenn der Trigger funktioniert. Wir verwenden für dieses Beispiel die folgende Zeichenfolge: „Trigger works ok.”
    - Definieren Sie die **Zusätzlichen Header**. Verwenden Sie das `Heading:Value`-Format.
5. Implementieren Sie die Konfiguration.
6. Öffnen Sie Ihr Entwicklerportal und gehen Sie zu Ihrer Seite **Cloud Trigger API v1**.
7. Klicken Sie auf **Try it**.
8. Fügen Sie `TestCloudTrigger` in das **triggerID**-Feld ein. Klicken Sie auf **Send**.
    - Die Antwortinhalte enthalten die Bestätigung: „Trigger works ok.”

3.8.2. Cloud-Trigger mit Ihrem vor Ort gehosteten Control Center implementieren

3.8.2.1. Cloud-Trigger in Automation Builder konfigurieren

Dieser Abschnitt beschreibt, wie Sie den Cloud-Trigger in Automation konfigurieren, das auf Ihrem lokalen Server ausgeführt wird.

1. Öffnen Sie Ihren Automation Builder. Stellen Sie sicher, dass der Automation Builder mit Ihrem Control Center verbunden ist. Wählen Sie zu diesem Zweck **Datei** > **Optionen** >**Control Center** und prüfen Sie, ob die URL-Adresse Ihres Control Center angegeben ist.

   > **ANMERKUNG**
   > Eine Verbindung zwischen Automation Builder und Control Center bedeutet außerdem, dass beide Anwendungen denselben Lizenzschlüssel nutzen.

2. Die Registerkarte **Konfigurationselemente** wird geöffnet. Klicken Sie auf **Cloud-Trigger**, um eine neue Konfiguration für den Cloud-Trigger zu erstellen.

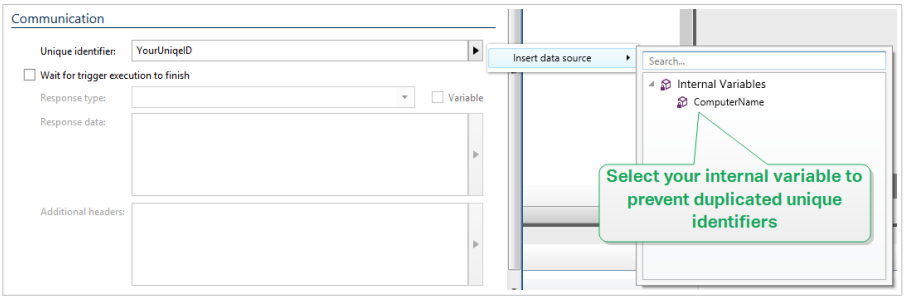
3. Stellen Sie **Namen** und **Beschreibung** ein, um Ihren Cloud-Trigger unter anderen Triggern leicht finden zu können.

4. Nehmen Sie die **Kommunikation**-Einstellungen für den Trigger vor:

   - Legen Sie die **Eindeutige Kennung** fest. Nachdem Sie den Trigger bereitgestellt haben, registriert diese eindeutige Kennung den Trigger in Ihrem Control Center. Verwenden Sie nur alphanumerische Zeichen. Sonderzeichen sind nicht erlaubt.
     Wenn Sie die Cloud-Trigger-Konfiguration auf mehreren Computern durchführen, müssen Sie sicherstellen, dass jeder Computer automatisch seine eigene eindeutige Kennung verwendet. Um unerwünschte Duplikate zu vermeiden, fügen Sie interne Variablen als Teil der **Eindeutigen Kennung** ein. Sie können zu diesem Zweck zwei interne Variablen verwenden:

     - **ComputerName**: Der Name des Computers, auf dem die Konfiguration ausgeführt wird.
     - **SystemUserName**: Der Windows-Benutzername des aktuell angemeldeten Benutzers.

     Um interne Variablen in die **Eindeutige Kennung** einzufügen, klicken Sie auf „Datenquelle einfügen“ und wählen Sie Ihre internen Variablen aus.



- **Warten, bis Trigger-Ausführung abgeschlossen ist:** Das HTTP-Protokoll wartet darauf, dass der Empfänger (in diesem Fall NiceLabel Automation) eine numerische Antwort an den Absender übermittelt, die den Status der empfangenen Nachricht anzeigt. Standardmäßig antwortet NiceLabel Automation mit dem Code 200. Dies zeigt an, dass Automation die Daten erfolgreich empfangen hat, gibt aber keine Informationen über den Erfolg von Trigger-Aktionen.
  Diese Option legt fest, dass ein Trigger nicht sofort nach Empfang der Daten eine Antwort sendet, sondern wartet, bis alle Aktionen ausgeführt wurden. Danach sendet er den Antwort-Code, um das erfolgreiche Ausführen von Aktionen zu bestätigen. Wenn diese Option aktiviert ist, können Sie eine benutzerdefinierte Antwort und die entsprechenden Daten zurücksenden (z. B. ist die Antwort auf eine HTTP-Abfrage die Etikettenvorschau im PDF-Format).
  Beim Cloud-Trigger sind die relevanten, in Automation integrierten Standard-HTTP-Antwortcodes:

HTTP-Antwortcode	Beschreibung
200	Alle Aktionen erfolgreich ausgeführt.
400	Keine Konfiguration verfügbar.
500	Beim Ausführen der Aktion sind Fehler aufgetreten.

ANMERKUNG

Um Feedback zum Druckprozess an Automation zu senden, aktivieren Sie den **synchronen** Druckmodus. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

- **Antworttyp:** Gibt den Typ Ihrer Antwortnachricht an. Häufig verwendete Internet-Medientypen (auch MIME-Typen oder Content-Typen genannt) stehen in der Dropdown-Liste zur Verfügung. Falls Ihr Medientyp nicht in der Liste enthalten ist, geben Sie ihn selbst ein. Automation sendet die ausgehenden Antwortdaten im Format des festgelegten Medientyps als Feedback. **Variable** aktiviert variable Medientypen. Wählen Sie bei aktivierter Option eine Variable aus (oder erstellen Sie eine Variable), die den Medientyp enthält.

ANMERKUNG

Wenn Sie keinen MIME-Typ angeben, verwendet NiceLabel Automation `application/octet-stream` als Standard.

- **Antwortdaten:** Definiert den Inhalt Ihrer Antwortnachricht. Beispiele für Inhalte, die Sie in Form von HTTP-Antworten senden können: Benutzerdefinierte Fehlermeldungen, Etikettenvorschau, erzeugte PDF-Dateien, Druckstrom-(Spool-)Dateien, XML-Dateien mit Daten aus der Druck-Engine und der Etikettenvorschau (als Base64-Zeichenfolge codiert) usw.
  Wenn Ihre Ausgabe aus binären Inhalten besteht (z. B. Etikettenvorschau oder Druckstrom), müssen Sie einen unterstützten Medientyp auswählen, z. B. `image/jpeg` oder `application/octet-stream`.

- **Zusätzliche Header**: Ermöglicht es Ihnen, benutzerdefinierte MIME-Header für die HTTP-Antwortnachricht zu definieren.
  Die Syntax für Antwortheader und Beispiele stehen im Abschnitt Aktion „HTTP-Anfrage“ zur Verfügung.

  TIPP
  Sie können bei **Antwortdaten** und **zusätzlichen Headern** festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzufügen, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie Ihre Variable aus der Liste ein. Sie können auch eine neue Variable erstellen, die die Daten enthält, sie Sie verwenden möchten. Weitere Informationen finden Sie im Abschnitt Zusammengesetzte Werte verwenden im Benutzerhandbuch.

5. Stellen Sie den Trigger in Automation Manager bereit und starten Sie ihn. Der Cloud-Trigger prüft jetzt auf eingehende Anfragen.

ANMERKUNG
Wenn Ihre Konfiguration höhere Verfügbarkeit und Skalierbarkeit erfordert, können Sie mehrere identische Cloud-Trigger bereitstellen. Installieren Sie zu diesem Zweck mehrere Instanzen von Automation und stellen Sie die Cloud-Trigger auf ihnen bereit. Wenn die bereitgestellten Cloud-Trigger dieselbe **Eindeutige Kennung** haben, verteilt der integrierte Lastausgleich in NiceLabel Cloud die Traffic-Last gleichmäßig auf sie.

3.8.2.2. Cloud-Trigger aufrufen (Vor-Ort-Implementierung)

Mit diesem Schritt stellen Sie sicher, dass die Ausgaben der externen Geschäftssysteme lokal gehostete Cloud-Trigger erfolgreich ausführen. Dies ist der Zweck der **CloudTrigger**Operation. Geben Sie in der URL für den Aufruf den Namen des Triggers an, den Sie aufrufen.

Um den Trigger mit Ihrer eindeutigen Kennung `MyCloudTrigger` aufzurufen, verwenden Sie die folgende URL:

```
https://<YourServerName>/epm/api/trigger/<MyCloudTriggerID>
```

ANMERKUNG
Die URL kann mit `http` oder `https` beginnen – je nachdem, wie Sie Ihr Control Center bei der Installation eingerichtet haben. Weitere Informationen finden Sie im Abschnitt „Website und Speicher einrichten“ des Control Center Installationshandbuchs.

Rufen Sie die URL für jedes Ereignis (Ausgabe) im externen Geschäftssystem wie im Beispiel gezeigt auf. Jeder Aufruf führt den Cloud-Trigger aus, der auf dem lokalen Automation Server gehostet wird.

Alle Aufrufe müssen einen Header namens **Integrator-Key** enthalten.

Beispiel

```
Integrator-Key: 9d59d7d444da412b8acfb488a01bb632
```

3.9. Planer-Trigger

Um mehr über Trigger im Allgemeinen zu erfahren, siehe Abschnitt Informationen zu Triggern.

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Der Planer-Trigger fungiert als Timer, der nach einer bestimmten Zeit mit der Ausführung von Aktionen in Ihrer Konfiguration beginnt. Nutzen Sie den Planer-Trigger, um eine wiederholte automatisierte Ausführung zeitabhängiger Aktionen einzurichten.

Der Planer-Trigger ist ein aktiver Trigger. Das bedeutet, dass er nicht auf ein Ereignis wartet, sondern mit der Ausführung der festgelegten Aktionen beginnt, sobald das angegebene Zeitintervall endet.

Beispiel

Typische Nutzung: Typischer Nutzungsfall: Ein ERP-System erstellt 6000 Verpackungs-Etikettendateien pro Tag. Automatisierung Druckt die Etiketten und speichert die verwendeten Etikettendateien in einem eigenen Verzeichnis. Aufgrund der großen Anzahl verwendeter Etikettendateien muss das Unternehmen ein automatisches Löschen von Dateien einrichten, die älter als 48 Stunden sind. Automatisierung nutzt den Planer-Trigger, um die nicht mehr benötigten Etikettendateien zu löschen.

Der Planer-Trigger berücksichtigt den Wechsel zwischen Sommer- und Winterzeit automatisch. Der Trigger nimmt immer die aktuelle Systemzeit als Referenz:

- Am Anfang der Sommerzeit führt der Trigger die jeweiligen Aktion zur ersten vollen Stunde nach dem übersprungenen Zeitpunkt aus.

  Beispiel

  Die Zeit für den Planer-Trigger ist auf 2.30 Uhr eingestellt. Die Uhr springt von 2 auf 3 Uhr. Der Trigger wird um 3 Uhr Sommerzeit ausgeführt.

- Am Ende der Sommerzeit wird der Trigger nur beim ersten Eintreten der festgelegten Zeit ausgeführt.

Beispiel

Die Zeit für den Planer-Trigger ist auf 2.30 Uhr eingestellt. Die Uhr springt von 3 auf 2 Uhr. Der Trigger wurde bereits ausgeführt. Daher führt er die Aktionen erneut aus, wenn die Uhr auf 2.30 Uhr Winterzeit steht.

3.9.1. Allgemein

In diesem Bereich können Sie die wichtigsten Trigger-Einstellungen vornehmen.

- **Name:** Gibt den eindeutigen Namen des Triggers an. Die Namen helfen Ihnen dabei, zwischen verschiedenen Triggern zu unterscheiden, wenn Sie sie in Automation Builder konfigurieren und später in Automation Manager ausführen.
- **Beschreibung:** Ermöglicht es Ihnen, die Rolle dieses Triggers zu beschreiben. Geben Sie für die Benutzer eine kurze Erklärung zur Funktionsweise des Triggers ein.

3.9.2. Erneutes Auftreten

Durch erneutes Auftreten können Sie festlegen, wie oft das Planer-Trigger-Ereignis wiederholt werden soll.

- **Trigger ausführen:** legt das Wiederholungsintervall für den Trigger fest.
  - **Alle (X) Sekunden/Minuten/Stunden** legt die Wiederholungszeit für das Trigger-Ereignis in den jeweiligen Zeiteinheiten fest. Legen Sie die Intervalldauer unter **Sekunden/Minuten/Stunden (X)** fest..
  - **Täglich** wiederholt das ausgewählte Trigger-Ereignis jeden Tag zur ausgewählten Uhrzeit. Legen Sie die Uhrzeit für die tägliche Wiederholung unter **Zeit:** fest.
  - **An bestimmten Tagen** wiederholt den Trigger zur festgelegten Uhrzeit am/an den ausgewählten Tag(en). Legen Sie die Wiederholung mithilfe von**Zeit:** und Tagen fest.

ANMERKUNG

Legen Sie die Zeitwerte im 24-Stunden-Format fest.

Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikationsparameter fest, durch die Sie Feedback von der Druck-Engine erhalten können.

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

- **Überwachtes Drucken:** Aktiviert den synchronen Druckmodus. Verwenden Sie ihn, wenn Sie den Status des Druckauftrags an die Drittanwendung zurücksenden wollen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Die Optionen im Abschnitt **Datenverarbeitung** geben an, ob Sie die Daten kürzen möchten, damit sie in eine Variable passen, oder ob die fehlenden Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet Fehler und unterbricht den Druckprozess, wenn Sie versuchen, zu lange Werte in Etikettenvariablen zu speichern oder Werte für nicht vorhandene Etikettenvariablen anzugeben.

- **Übermäßig lange Variableninhalte ignorieren:** kürzt Datenwerte, die die Länge der Variable gemäß Definition im Etiketten-Designer überschreiten, um sie auf passende Länge zu bringen. Diese Option wird wirksam, wenn Sie Variablenwerte in Filtern oder aus Befehlsdateien festlegen und wenn Sie Werte von Triggervariablen den gleichnamigen Etikettenvariablen zuordnen.

Beispiel

Die Etikettenvariable akzeptiert maximal 5 Zeichen. Wenn diese Option aktiviert ist, wird jeder Wert über 5 Zeichen auf die ersten 5 Zeichen gekürzt. Wenn der Wert 1234567 lautet, ignoriert die 6. Und 7. Stelle.

- **Fehlende Etikettenvariablen ignorieren:** Beim Drucken mit Befehlsdateien (z. B. JOB-Dateien), werden alle Variablen ignoriert, die:
  - in der Befehlsdatei angegeben sind (anhand des SET Befehls)
  - nicht auf dem Etikett definiert sind

  Etwas Ähnliches geschieht, wenn Sie einen Zuweisungsbereich in einem Filter definieren, um alle Name-Wert-Paare zu extrahieren, Ihr Etikett jedoch weniger Variablen enthält.
  Wenn Sie Werte von nicht vorhandenen Etikettenvariablen einstellen, gibt einen Fehler aus. Wenn diese Option aktiviert ist, wird der Druck fortgesetzt.

Die Optionen im Abschnitt **Scripting** geben die Scripting-Möglichkeiten an.

- **Scripting-Sprache:** Wählt die Scripting-Sprache für den Trigger aus. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden die ausgewählte Scripting-Sprache.

Die Optionen im Abschnitt **Empfangene Daten speichern** legen die Befehle fest, die für vom Trigger empfangene Daten zur Verfügung stehen.

- **Vom Trigger empfangene Daten speichern in Datei:** Aktivieren Sie diese Option, um vom Trigger empfangene Daten zu speichern. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und den Dateinamen enthält.
- **Bei Fehler vom Trigger empfangene Daten speichern in Datei:** Aktivieren sie diese Option, um die vom Trigger empfangenen Daten nur dann zu speichern, wenn ein Fehler bei der Ausführung der Aktion auftritt. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu behalten, die das Problem hervorgerufen haben, um die Lösungsfindung zu erleichtern.

ANMERKUNG

Stellen Sie sicher, dass Sie die Unterstützung für Überwachtes Drucken aktivieren. Ist sie nicht aktiviert, kann Fehler bei der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

ANMERKUNG

speichert die empfangenen Daten in einer temporären Datei. Diese temporäre Datei wird sofort nach Abschluss der Trigger-Ausführung gelöscht. Die interne Variable **`DataFileName`** verweist auf diesen Dateinamen. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

Sicherheit

- **Trigger sperren und verschlüsseln:** Aktiviert den Trigger-Schutz. Wenn Sie die Option aktivieren, wird der Trigger gesperrt und kann nicht mehr bearbeitet werden. Dadurch werden die Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

3.10. Trigger testen

Nach Einrichtung der Trigger ist die Konfiguration erst zur Hälfte abgeschlossen. Bevor Sie den Trigger implementieren, sollten Sie ihn durch Anwendung auf eingehende Daten sorgfältig testen, um sicherzustellen, dass er zum gewünschten Ergebnis führt.

Automation Builder ermöglicht es Ihnen, die Konfiguration zu testen, während Sie noch an ihr arbeiten. Einige Aktionen bieten integrierte Testmöglichkeiten, sodass Sie sich auf die Ausführung der jeweiligen Aktion konzentrieren können. Außerdem können Sie Trigger mit dem Befehl „Vorschau ausführen“ testen. Der finale Test sollte jedoch immer in der realen Umgebung anhand von echten Daten und echten Triggern erfolgen. So überwachen Sie die Trigger-Ausführung mit Automation Manager.

Ausführung einzelner Aktionen testen

Einige der Aktionen enthalten die Vorschauoption. Auf diese Weise können Sie die Eingabeparameter testen und das Ergebnis der Aktion auf dem Bildschirm sehen.

- **Datenfilter verwenden:** Die Aktion bietet eine Echtzeitvorschau der geparsten Daten. Die Regeln in dem ausgewählten Filter werden auf die ausgewählten Eingabedaten angewandt. Das Ergebnis wird in der Tabelle angezeigt. Wenn Sie Unter- oder Zuweisungsbereiche verwenden, können Sie die Vorschau für jede Ebene der Filterdefinition anzeigen.
- **SQL-Anweisung ausführen:** Mit dieser Aktion können Sie eine Vorschau der Ausführung der definierten SQL-Anweisung anzeigen. So überwachen Sie das aus der SELECT-Anweisung resultierende Daten-Set sowie die Anzahl von Zeilen, auf die sich die UPDATE-, INSERT- und DELETE-Anweisungen auswirken. Die Ausführung der Vorschau ist transaktionssicher, sodass Sie alle Änderungen rückgängig machen können. Sie können die Abfrageparameter ändern und erkennen, inwiefern sich die Änderung auf das Ergebnis auswirkt.
- **Webdienst:** Die Aktion zeigt eine Vorschau der Ausführung einer ausgewählten Methode (Funktion) vom Webdienst an. Sie können die Eingabeparameter ändern und erkennen, inwiefern sich die Änderung auf das Ergebnis auswirkt.
- **Script ausführen:** Die Aktion prüft das angegebene Skript auf Syntaxfehler und führt es zudem aus. Sie können die Eingabeparameter ändern und erkennen, inwiefern sich die Änderung auf die Skript-Ausführung auswirkt.

Ausführung des Triggers testen und Vorschau auf dem Bildschirm anzeigen

Um den Trigger von Grund auf zu testen, verwenden Sie die integrierte Funktion **Vorschau ausführen**. Sie können eine Vorschau für alle Trigger-Typen ausführen. Der Trigger löst bei Veränderungen im überwachten Ereignis nicht aus. Dies ist nur bei einem im Automation Manager gestarteten Trigger der Fall. Stattdessen wird der Trigger Aktionen auf Basis von in einer Datei gespeicherten Daten ausführen. Stellen Sie sicher, dass Sie eine Datei mit Beispieldaten verwenden, welche der Trigger in einer Echtzeit-Implementierung akzeptieren würde.

Der Trigger führt alle definierten Aktionen aus, einschließlich Datenfilterung, und zeigt die Etikettenvorschau auf dem Bildschirm an. Die Vorschau simuliert den Druckprozess bis ins kleinste Detail. Die Etiketten würden mit derselben Zusammensetzung und denselben Inhalten gedruckt, die in der Vorschau angezeigt werden. Dies schließt die Anzahl von Etiketten sowie deren Inhalte mit ein. So erkennen Sie auch, wie viele Druckaufträge erzeugt werden, wie viele Etiketten in jedem Auftrag enthalten sind, und Sie erhalten eine Vorschau aller einzelnen Etiketten. In einem ausgewählten Druckauftrag können Sie zwischen den einzelnen Etiketten wechseln.

Der Protokollbereich zeigt dieselben Informationen an, die auch im Automation Manager angezeigt würden. Protokolleinträge erweitern, um alle Details zu sehen.

ANMERKUNG

Nach Ausführen der Vorschau werden alle für den ausgewählten Trigger definierten Aktionen ausgeführt, nicht nur die Aktion „Vorschau ausführen“. Seien Sie vorsichtig, wenn Sie Aktionen verwenden, die zu einer Änderung der Daten führen, z. B. SQL-Anweisung ausführen oder Webdienst, da ihre Ausführung nicht rückgängig gemacht werden kann.

Um eine Vorschau der Etiketten anzuzeigen, tun Sie Folgendes:

1. Öffnen Sie die Trigger-Konfiguration.
2. Stellen Sie sicher, dass die Trigger-Konfiguration gespeichert wurde.
3. Klicken Sie auf die Schaltfläche **Vorschau ausführen** in der Vorschau-Gruppe der Multifunktionsleiste.
4. Navigieren Sie zu der Datendatei, die die typischen, vom Trigger akzeptierten Inhalte bereitstellt.
5. Sehen Sie sich das Ergebnis auf der Registerkarte „Vorschau“ an.

Implementierung auf dem Vorproduktions-Server testen

Es empfiehlt sich, die Konfiguration auf einem Vorproduktions-Server in Automation Manager zu implementieren, bevor die Implementierung auf dem Produktions-Server durchgeführt wird. Beim Testen in einer Vorproduktionsumgebung könnten weitere Konfigurationsprobleme erkannt werden, die im Rahmen der Trigger-Tests im Automation Builder nicht erkannt wurden.

Außerdem kann ein Belastungstest durchgeführt werden, indem die Last auf den Trigger verstärkt und dessen Performance geprüft wird. Die Tests bieten wichtige Informationen zum verfügbaren Durchsatz und zu potenziellen Schwachpunkten. Auf Basis dieser Informationen können Sie verschiedene Systemoptimierungstechniken anwenden, beispielsweise eine Optimierung des Etikettendesigns zur Erstellung kleinerer Druckdatenströme und eine Optimierung des gesamten Datenflusses aus der vorhandenen Anwendung in NiceLabel Automation.

Wichtige Unterschiede zwischen Trigger-Tests unter realen Bedingungen und der Vorschau in Automation Builder

Die Vorschau des Triggers in Automation Builder stellt eine schnelle Möglichkeit zum Testen des Triggers dar, aber Sie dürfen sich nicht allein darauf verlassen. Es kann Unterschiede zwischen der Vorschau und der Ausführung des Triggers in einer realen Umgebung unter Nutzung von 64-Bit-Windows geben.

Selbst wenn Ihre Konfiguration in Automation Builder einwandfrei funktioniert, sollten Sie sie auf jeden Fall auch unter realen Bedingungen anhand des Dienstes testen.

- Wenn Sie den Befehl **Vorschau ausführen** verwenden, wird die Konfiguration in Automation Builder ausgeführt, das immer als 32-Bit-Anwendung läuft. Die Vorschau Ihres Triggers in Automation Builder testet also nur die Ausführung auf einer 32-Bit-Plattform.
- Wenn Sie Trigger unter realen Bedingungen testen, wird die Konfiguration im Dienst ausgeführt, welcher unter 32-Bit-Windows als 32-Bit-Anwendung und unter 64-Bit-Windows als 64-Bit-Anwendung läuft. Weitere Informationen finden Sie im Abschnitt Im Dienstmodus ausführen.
- Es kann zu Problemen kommen, wenn sich Plattformunterschiede (32 Bit vs. 64 Bit) auf die Trigger-Verarbeitung auswirken:
  - **Datenbankzugriff:** 64-Bit-Anwendungen erfordern 64-Bit-Datenbanktreiber, während 32-Bit-Anwendungen 32-Bit-Datenbanktreiber benötigten. Um die Konfiguration aus Automation Builder und im Dienst auszuführen, benötigen Sie 32- und 64-Bit-Treiber für den Zugriff auf Ihre Datenbank. Weitere Informationen finden Sie im Abschnitt Zugriff auf Datenbanken.
  - **UNC-Syntax für Netzwerkdateien:** Das Dienstkonto kann nicht auf im Netzwerk freigegebene Dateien mit zugeordnetem Laufwerksbuchstaben zugreifen. Sie müssen UNC-Syntax für Dateien im Netzwerk verwenden. Verwenden Sie zum Beispiel `\\server\share\files\label.lbl` anstelle von `G:\files\label.lbl`, wobei „G:“ `\\server\share` zugeordnet ist. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk.
- Falls Ihr NiceLabel Automation-Dienst nicht unter dem Benutzeraccount ausgeführt wird, den Sie für Automation Builder verwenden, haben die Konten eventuell nicht dieselben Zugriffsrechte. Auch wenn Sie das Etikett in Automation Builder öffnen können, kann das Benutzerkonto für den Dienst möglicherweise nicht darauf zugreifen. Um Automation Builder unter demselben Benutzerkonto wie den Dienst auszuführen, siehe Dasselbe Benutzerkonto zur Konfiguration und Ausführung von Triggern verwenden.

3.11. Sicheres Übertragungsprotokoll (HTTPS) nutzen

PRODUKTEBENEN-INFO

Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

Sie können den beim HTTP Server Trigger und beim Webdienst-Triggereingehenden Datenverkehr schützen, indem Sie die HTTPS-Unterstützung aktivieren. HTTPS sichert die Übertragung von Nachrichten über das Netzwerk. Hierbei werden X.509-Zertifikate verwendet, um die zwischen den Parteien übertragenen Daten zu verschlüsseln. Ihre Informationen bleiben vor unautorisiertem Zugriff geschützt, da nur der Client und NiceLabel Automation den Datenverkehr entschlüsseln können. Selbst wenn ein unautorisierter Benutzer versuchen würde, die Kommunikation abzuhören, könnte er die Bedeutung der Nachrichten nicht verstehen, da der Datenverkehr als Strom von zufälligen Bytes erscheint.

Unter bestimmten Umständen ist es ratsam, die Kommunikation zu verschlüsseln, zum Beispiel wenn:

- Sie mit sensiblen und vertraulichen Daten arbeiten, die Dritten nicht zugänglich gemacht werden dürfen.
- Die Nachrichten Netzwerke durchlaufen müssen, über die Sie keine Kontrolle haben. Dies ist beispielsweise der Fall, wenn Sie Daten über das Internet statt über das lokale Netzwerk an Automation senden.

Das sichere Übertragungsprotokoll (HTTPS) aktivieren

So aktivieren Sie das sichere Übertragungsprotokoll für Ihren Trigger.

Unter Windows:

1. Fordern Sie das X.509-Zertifikat von der Zertifizierungsstelle (CA) an. Sie benötigen einen Zertifikatstyp für „Server-Authentifizierung“.

   > **ANMERKUNG**
   > Wenn Sie das Zertifikat selbst generieren wollen, müssen Sie das CA-Zertifikat zum Speicher vertrauenswürdiger Zertifizierungsstellen hinzufügen, damit die CA-Signatur auf dem Server-Zertifikat verifiziert werden kann.

2. Installieren Sie das X.509-Zertifikat an dem Speicherort im System, wo NiceLabel Automation installiert ist. Stellen Sie sicher, dass das Zertifikat für das Benutzerkonto sichtbar ist, unter dem Sie den NiceLabel Automation-Dienst ausführen. Es empfiehlt sich, das Zertifikat im Speicher des lokalen Computers zu installieren, nicht im Speicherbereich des aktuellen Benutzers. So kann NiceLabel Automation das Zertifikat auch dann nutzen, wenn es nicht unter Ihrem aktuell angemeldeten Benutzerkonto ausgeführt wird.

   a. Öffnen Sie ein Eingabeaufforderungs-Fenster.

   b. Geben Sie **mmc** ein und drücken Sie die Eingabetaste (stellen Sie sicher, dass Sie die Anwendung mit Administratorrechten ausführen).

   c. Klicken Sie im Datei-Menü auf **Snap-In hinzufügen/entfernen**.

   d. Wählen Sie im Dialogfeld **Eigenständiges Snap-In hinzufügen** die Option **Zertifikate**.

   e. Klicken Sie auf **Hinzufügen**.

   f. Wählen Sie im Dialogfeld **Zertifikat-Snap-In** die Option **Computerkonto** und klicken Sie auf **Weiter**.

   g. Klicken Sie im Dialogfeld **Computer auswählen** auf **Fertig stellen**.

   h. Klicken Sie im Dialogfeld **Snap-In hinzufügen/entfernen** auf **OK**.

   i. Klicken Sie im Konsolenstamm-Fenster **Zertifikate>Persönlich**.

   j. Klicken Sie mit der rechten Maustaste auf das Zertifikate-Verzeichnis und wählen Sie **Alle Aufgaben>Importieren**.

   k. Folgen Sie den Schritten im Assistenten, um das Zertifikat zu importieren.

3. Rufen Sie den Fingerabdruck des Zertifikats ab, das Sie soeben importiert haben.

   a. Doppelklicken Sie bei geöffneter mmc-Konsole auf das Zertifikat.

   b. Klicken Sie im **Zertifikat**-Dialogfeld auf die **Details**-Registerkarte.

   c. Scrollen Sie durch die Liste von Feldern und klicken Sie auf „Fingerabdruck“.

d. Kopieren Sie die hexadezimalen Zeichen aus dem Feld. Entfernen Sie die Leerzeichen zwischen den Hexadezimalzahlen. Der Fingerabdruck „a9 09 50 2d d8 2a e4 14 33 e6 f8 38 86 b0 0d 42 77 a3 2a 7b“ sollte beispielsweise im Code als „a909502dd82ae41433e6f83886b00d4277a32a7b“ angegeben werden. Dies ist der im nächsten Schritt erforderliche **`certhash`**.

4. Binden Sie das Zertifikat an die IP-Adresse und den Port, an denen der Trigger ausgeführt wird. Dadurch wird das Zertifikat an der ausgewählten Portnummer aktiviert.
   Öffnen Sie die **Eingabeaufforderung** (Sie müssen sie mit Administratorrechten ausführen) und führen Sie den folgenden Befehl aus:

```
netsh http add sslcert ipport=0.0.0.0:56000
certhash=7866c25377554ca0cb53bcdfd5ee23ce895bdfa2
appid={A6BF8805-1D22-42C2-9D74-3366EA463245}
```

   wobei:

   - **`ipport`** das IP-Adresse/Port-Paar ist, an dem der Trigger ausgeführt wird. Behalten Sie für die IP-Adresse 0.0.0.0 (lokaler Computer) bei, aber ändern Sie die Portnummer, sodass sie der Portnummer in der Trigger-Konfiguration entspricht.
   - **`certhash`** ist der Fingerabdruck (SHA-Hash) des Zertifikats. Er hat eine Länge von 20 Byte und ist als hexadezimale Zeichenfolge angegeben.
   - **`appid`** die GUID der jeweiligen Anwendung ist. Sie können hier jede GUID verwenden, auch die aus dem obigen Beispiel.

Führen Sie folgende Schritte in der Trigger-Konfiguration aus:

1. Aktivieren Sie in Ihrem HTTP- oder Webdienst-Trigger die Option **Sichere Verbindung (HTTPS)**.
2. Laden Sie die Konfiguration im Automation Manager neu.

Das sichere Übertragungsprotokoll (HTTPS) deaktivieren
Unter Windows:

- Heben Sie die Bindung zwischen dem Zertifikat und dem IP-Adresse/Port-Paar auf. Führen Sie den folgenden Befehl in der Eingabeaufforderung (Sie müssen sie mit Administratorrechten ausführen) aus:

```
netsh http delete sslcert ipport=0.0.0.0:56000
```

  wobei:

  - **`ipport`** ist das IP-Adressen-Portpaar, an dem der Trigger ausgeführt wird und mit dem Sie das Zertifikat verbunden haben.

Führen Sie folgende Schritte in der Trigger-Konfiguration aus:

1. Deaktivieren Sie in Ihrem HTTP- oder Webdienst-Trigger die Option **Sichere Verbindung (HTTPS)**.
2. Laden Sie die Konfiguration im Automation Manager neu.

3.12. Trigger-Konfiguration vor Bearbeitung schützen

Die Trigger-Konfiguration kann auf zwei Arten geschützt werden.

- **Trigger sperren.** Anhand dieser Methode sperren Sie die Trigger-Konfigurationsdatei und schützen sie mit einem Passwort. Ohne das Passwort kann niemand den Trigger bearbeiten. Aktivieren Sie die Option **Trigger sperren und verschlüsseln** unter *Einstellungen -> Sicherheit*.
- **Zugriffsrechte einstellen.** Anhand dieser Methode können Sie die in den NiceLabel Automation-Optionen festgelegten Benutzerrechte zum Schutz der Trigger-Konfiguration nutzen. Sie können Benutzergruppen aktivieren und jeder Gruppe unterschiedliche Rollen zuweisen. Sind der Gruppe Bearbeitungsrechte zugewiesen, können alle Mitglieder Trigger bearbeiten. Um diese Methode zu verwenden, müssen Sie die Benutzeranmeldung aktivieren. Sie können Windows-Benutzer aus lokalen Gruppen oder Active Directory nutzen oder NiceLabel Automation-Benutzer definieren. Siehe **Benutzerrechte und Zugriff** im Abschnitt zur Konfiguration.

3.13. Firewall für Netzwerk-Trigger konfigurieren

Ein Netzwerk-Trigger ist ein Trigger, der anhand des TCP/IP-Protokolls ausgeführt wird. In Automation sind solche Trigger TCP/IP-Trigger, HTTP-Trigger und Webdienst-Trigger. Sie stellen Netzwerkdienste bereit und sind an die Netzwerkschnittstellenkarte, ihre IP-Adresse und die konfigurierte Portnummer gebunden. Nach der Implementierung und dem Starten von Netzwerk-Triggern in Automation Manager beginnen sie mit der Überwachung des Ports für eingehenden Datenverkehr.

Firewalls schützen Computer vor unbefugten eingehenden Verbindungsversuchen. Das NiceLabel Installationsprogramm stellt sicher, dass eingehende Kommunikationsströme an alle vom Automation Dienst belegten Ports von der Windows Firewall zugelassen werden.

> **WARNUNG**
> Der Automation Dienst belegt Ports, die für TCP/IP-Trigger konfiguriert sind, aber keine Ports für HTTP- und Webdienst-Trigger. Diese Ports sind an den ID 4 (SYSTEM) Prozess gebunden, nicht an den AutomationDienstprozess.

Konfigurieren Sie die Firewall, um die Kommunikation an Ports zu erlauben, die für HTTP- und Webdienst-Trigger konfiguriert sind. Um eine Regel für eingehende Daten zu erstellen, tun Sie Folgendes:

1. Öffnen Sie auf dem Computer, auf dem NiceLabel Automation ausgeführt wird, im **Start**-Menü die **Systemsteuerung**, wählen Sie **System und Sicherheit** und dann **Windows Firewall**.
2. Wählen Sie im Navigationsbereich die Option **Erweiterte Einstellungen**.
3. Wählen Sie im Fenster **Windows-Firewall mit erweiterter Sicherheit** im Navigationsbereich die Option **Eingehende Regeln** und dann im Aktionen-Bereich die Option **Neue Regel**.
4. Wählen Sie auf der Seite **Regeltyp** die Option **Port** und klicken Sie auf **Weiter**.
5. Wählen Sie auf der Seite **Protokoll und Ports** die Option **Bestimmte lokale Ports** und geben Sie die Portnummer ein, an der Ihr HTTP- oder Webdienst-Trigger ausgeführt wird.
6. Klicken Sie auf **Weiter**.
7. Wählen Sie auf der Seite **Aktionen** die Option **Verbindung zulassen** und klicken Sie auf **Weiter**.
8. Wählen Sie auf der Seite **Profil** die Profile aus und klicken Sie auf **Weiter**.
9. Geben Sie auf der Seite **Name** einen Namen für die Regel ein und klicken Sie auf **Fertig stellen**.

Bei anderer Firewall-Software müssen Sie ähnliche Schritte ausführen.

4. Variablen

Variablen werden als Behälter für Datenwerte verwendet. Sie benötigen Variablen, um Werte für die Aktion **Etikett drucken** an das Etikett zu übermitteln oder um Werte bei anderen Datenmanipulations-Aktionen zu verwenden. Für gewöhnlich extrahiert ein Filter Werte aus vom Trigger empfangenen Datenströmen und sendet diese an Variablen. Weitere Informationen finden Sie im Abschnitt Informationen zu Filtern.

Normalerweise senden Sie die Werte von Variablen an eine zu druckende Etikettenvorlage. Der Mechanismus, der Werte von Variablen an Etiketten sendet, nutzt die automatische Namenszuordnung. Der Wert der im Trigger definierten Variablen wird an die im Etikett definierte Variable mit demselben Namen gesendet. Sie können Variablen auf eine von drei verschiedenen verfügbaren Arten definieren:

- **Variablen aus Etikettendatei importieren:** Wie für die oben beschriebene automatische Zuordnung empfiehlt es sich, Ihre Variablen jedes Mal vom Etikett zu importieren. Diese Aktion spart Ihnen Zeit und stellt sicher, dass die Variablennamen übereinstimmen. Die importierte Variable übernimmt nicht nur den Variablennamen, sondern auch unterstützte Variableneigenschaften wie Länge und Standardwert.
- **Variablen manuell definieren:** Wenn Sie Variablen manuell definieren, müssen Sie darauf achten, dieselben Namen zu verwenden, die auch die Variablen im Etikett tragen. Manuell definieren müssen Sie insbesondere diejenigen Variablen, die im Etikett nicht vorhanden sind, die Sie jedoch im Trigger benötigen.

  > **ANMERKUNG**
  > Beispiele dafür sind Variablen wie **`LabelName`**, **`PrinterName`**, **`Quantity`** und ähnliche Variablen, die Sie benötigen, um den Etikettennamen, den Druckernamen, die Menge oder andere vom Filter zugewiesene Metadaten zu speichern.

- **Interne Variablen aktivieren:** Werte für interne Variablen werden von NiceLabel Automation zugewiesen und stehen als Nur-Lesen-Werte zur Verfügung. Weitere Informationen finden Sie in den Abschnitten Interne Variablen.

> **TIPP**
> Wenn Sie den **Zuweisungsbereich** (im Filter für unstrukturierten Text und im XML-Filter) oder die **dynamische Struktur** (im Filter für strukturierten Text) aktivieren, extrahiert NiceLabel Automation **Name:Wert**-Paare aus den Trigger-Daten und sendet die Werte automatisch an die gleichnamigen Variablen, die im Etikett definiert sind. Eine manuelle Variablenzuordnung ist nicht notwendig.

Eigenschaften

- **Name:** Gibt den eindeutigen Namen der Variablen an. Bei Namen wird Groß-/Kleinschreibung nicht beachtet. Obwohl Leerzeichen in Variablennamen verwendet werden können, ist davon abzuraten. Dies gilt umso mehr, wenn Sie Variablen in Skripten oder in Bedingungen für Aktionen verwenden, da sie in diesen Fällen in eckigen Klammern eingeschlossen werden müssen.
- **Erlaubte Zeichen:** Legt die Liste von Zeichen fest, die ein Wert beinhalten darf. Sie können zwischen **Alle** (alle Zeichen werden akzeptiert), „Numerisch“ (nur Ziffern werden akzeptiert) und „Binär“ (alle Zeichen und Binärcodes werden akzeptiert).
- **Variablenlänge begrenzen:** Legt die maximale Anzahl von Zeichen fest, die eine Variable enthalten darf.
- **Feste Länge:** Legt fest, dass der Wert exakt aus einer definierten Anzahl von Zeichen bestehen muss.

  ANMERKUNG
  Für bestimmte Etikettenobjekte muss die Variablenlänge begrenzt werden. Ein Beispiel ist ein EAN-13-Barcode, der aus 13 Zeichen besteht.
- **Wert erforderlich:** Legt fest, dass die Variable einen Wert enthalten muss.
- **Standardwert:** Legt einen Standardwert fest. Wenn der Variablen kein Wert zugewiesen ist, wird immer der Standardwert verwendet.

4.1. Zusammengesetzte Werte verwenden

Einige Objekte in der Trigger-Konfiguration akzeptieren zusammengesetzte Werte. Solche Inhalte können aus einer Kombination von festen Werten, variablen Werten und Sonderzeichen (Steuerzeichen) bestehen. Die Objekte, die zusammengesetzte Werte akzeptieren, sind durch eine kleine Pfeil-Schaltfläche an ihrer rechten Seite gekennzeichnet. Klicken Sie auf diese Schaltfläche, um entweder eine Variable oder ein Sonderzeichen einzufügen.

- **Feste Werte verwenden:** Geben Sie manuell einen festen Wert für die Variable ein.

  ```
  This is fixed value.
  ```

- **Feste Werte und Daten aus Variablen verwenden:** Sie können einen zusammengesetzten Wert definieren, der aus variablen und festen Werten besteht. Die Variablennamen müssen in eckige Klammern **[]** eingeschlossen werden. Sie können Variablen manuell eingeben oder durch Klicken auf die Pfeil-Schaltfläche auf der rechten Seite einfügen. Zum Zeitpunkt der Verarbeitung werden die Werte von Variablen mit festen Daten verbunden und als Inhalt verwendet. Weitere Informationen finden Sie im Abschnitt Tipps und Tricks zur Nutzung von Variablen in Aktionen.
  In diesem Fall besteht der Inhalt aus drei Variablen und manuell eingegebenen Festdaten.

  ```
  [variable1] // This is fixed value [variable2][variable3]
  ```

- **Sonderzeichen verwenden:** Sie können Sonderzeichen zu der Kombination hinzufügen. Geben Sie die Sonderzeichen manuell ein oder fügen Sie sie ein. Weitere Informationen finden Sie im Abschnitt Eingabe von Sonderzeichen.
  Im folgenden Fall wird der Wert von **`Variable1`** mit festen Daten und dem binären Steuerzeichen „Form Feed“ verbunden.

```
[variable1] Form feed will follow this fixed text <FF>
```

4.2. Interne Variablen

Interne Variablen werden von NiceLabel Automation vorgegeben. Ihre Werte werden automatisch zugewiesen und stehen als Nur-Lesen-Werte zur Verfügung. Das Schloss-Symbol vor dem Variablennamen unterscheidet interne Variablen von benutzerdefinierten Variablen. Sie können interne Variablen in Ihren Aktionen auf dieselbe Weise verwenden wie benutzerdefinierte Variablen. Die Trigger-bezogenen internen Variablen funktionieren für jeden Trigger als interne Variablen.

Interne Variable	Verfügbar in Trigger	Beschreibung
ActionLastErrorDesc	Alle	Stellt die Beschreibung des zuletzt aufgetretenen Fehlers bereit. Verwenden Sie diesen Wert als Feedback für das Host-System, um die Ursache des Fehlers zu identifizieren.
ActionLastErrorID	Alle	Stellt die letzte Fehlerkennung bereit. Dies ist ein Ganzzahlwert. Wenn der Wert 0 beträgt, gab es keinen Fehler. Sie können diesen Wert in Bedingungen nutzen, um zu prüfen, ob ein Fehler aufgetreten ist oder nicht.
BytesOfReceivedData	TCP/IP	Stellt die Anzahl der vom Trigger empfangenen Bytes bereit.
ComputerName	Alle	Stellt den Namen des Computers bereit, auf dem die Konfiguration ausgeführt wird.
ConfigurationFileName	Alle	Stellt den Pfad und den Dateinamen der aktuellen Konfiguration bereit (.MISX-Datei).
ConfigurationFilePath	Alle	Stellt den Pfad der aktuellen Konfigurationsdatei bereit. Siehe auch Beschreibung für „ConfigurationFileName“.
DataFileName	Alle	Stellt den Pfad und den Dateinamen der Arbeitskopie der empfangenen Daten bereit. Jedes Mal, wenn der Trigger Daten empfängt, wird eine Backup-Kopie von ihnen unter dem eindeutigen Dateinamen erstellt, der von dieser Variablen angegeben wird.

Database	Database	Stellt den Datenbank-Typ entsprechend der Trigger-Konfiguration bereit.
Date	Alle	Stellt das aktuelle Datum im Format bereit, das durch das Systemgebietsschema vorgegeben wird, z. B. „26.2.2018“.
DateDay	Alle	Stellt den aktuellen Tag im Monat bereit, z. B. „26“.
DateMonth	Alle	Stellt den aktuellen Monat im Jahr bereit, z. B. „2“.
DateYear	Alle	Stellt das aktuelle Jahr bereit, z. B. „2018“.
DefaultPrinterName	Alle	Stellt den Namen des als Standard definierten Druckers bereit.
DriverType	Database	Stellt den Namen des Treibers bereit, der zur Verbindung mit der ausgewählten Datenbank verwendet wird.
Hostname	TCP/IP	Stellt den Hostnamen des Geräts/Computers bereit, das/der sich mit dem Trigger verbindet.
HttpMethodName	HTTP	Stellt den Methodennamen bereit, den der Benutzer in der HTTP-Anfrage angegeben hat, z. B. GET oder POST.
HttpPath	HTTP	Stellt den im HTTP-Trigger definierten Pfad bereit.
HttpQuery	HTTP	Stellt den Inhalt der vom HTTP-Trigger empfangenen Abfragefolge bereit.
NumberOfRowsReturned	Database	Stellt die Anzahl der Zeilen bereit, die der Trigger von einer Datenbank erhält.
LocalIP	TCP/IP	Stellt die lokale IP-Adresse bereit, an der der Trigger antwortet. Dies ist nützlich, wenn Sie einen Multi-Homing-Rechner mit mehreren Netzwerkkarten (NIC) haben und bestimmen wollen, mit welcher IP-Adresse sich der Client verbunden hat. Dies ist beispielsweise beim Austausch von Druckern nützlich.
PathDataFileName	Alle	Stellt den Pfad in der Variablen „DataFileName“ ohne den Dateinamen bereit. Siehe auch Beschreibung für „DataFileName“.
PathTriggerFileName	Datei	Stellt den Pfad in der Variablen „TriggerFileName“ ohne den Dateinamen bereit. Siehe auch Beschreibung für „TriggerFileName“.

Portname	TCP/IP, HTTP, Webdienst	Stellt die Schnittstellennummer gemäß Definition im Trigger bereit.
RemoteHttpIp	HTTP	Stellt den Hostnamen des Geräts/Computers bereit, das/der sich mit dem Trigger verbindet.
RemoteIp	Webdienst	Stellt den Hostnamen des Geräts/Computers bereit, das/der sich mit dem Trigger verbindet.
ShortConfigurationFileName	Alle	Stellt den Namen der Konfigurationsdatei ohne Pfad bereit. Siehe auch Beschreibung für „ConfigurationFileName“.
ShortDataFileName	Alle	Stellt den Dateinamen in der Variablen „DataFileName“ ohne den Pfad bereit. Siehe auch Beschreibung für „DataFileName“.
ShortTriggerFileName	Datei	Stellt den Dateinamen in der Variablen „TriggerFileName“ ohne den Pfad bereit. Siehe auch Beschreibung für „TriggerFileName“.
SystemUserName	Alle	Stellt den Windows-Namen des angemeldeten Benutzers bereit.
TableName	Database	Stellt den Namen der im Trigger verwendeten Tabelle bereit.
Time	Alle	Stellt die aktuelle Zeit im Format bereit, das durch das Systemgebietsschema vorgegeben wird, z. B. „15:18“.
TimeHour	Alle	Stellt die aktuelle Stunde bereit, z. B. „15“.
TimeMinute	Alle	Stellt die aktuelle Minute bereit, z. B. „18“.
TimeSecond	Alle	Stellt die aktuelle Sekunde bereit, z. B. „25“.
TriggerFileName	Datei	Stellt den Namen der Datei bereit, die Aktionen ausgelöst hat. Dies ist nützlich, wenn Sie eine Reihe von Dateien in einem Ordner überwachen und feststellen wollen, welche Datei die Aktionen ausgelöst hat.
TriggerName	Alle	Stellt den Namen des Triggers gemäß Definition durch den Benutzer bereit.
Username	Alle	Stellt den NiceLabel Automation-Benutzernamen des aktuell angemeldeten Benutzers bereit. Die Variable hat nur dann einen Inhalt, wenn die Benutzeranmeldung aktiviert ist.

4.3. Globale Variablen

Globale Variablen sind Variablen, die auf mehreren Etiketten verwendet werden können. Sie werden außerhalb der Etikettendatei definiert und speichern die zuletzt genutzten Werte.

Globale Variablen werden für gewöhnlich als globale Zähler definiert. Solche globalen Variablen stellen einen eindeutigen Wert für jedes Etikett bereit, das einen neuen Wert anfordert. Durch eine Dateisperre wird sichergestellt, dass jeder Wert eindeutig ist.

Globale Variablen werden im Etiketten-Designer definiert; NiceLabel Automation greift nur auf sie zu. Die Quelle für globale Variablen lässt sich im **Optionen**-Dialog (**Datei** > **Optionen** > **Globale Variablen**) konfigurieren.

Standardmäßig ist NiceLabel Automation für die Nutzung globaler Variablen vom lokalen Computer konfiguriert. Ihr Standard-Speicherort ist:

```
%PROGRAMDATA%\NiceLabel\Global Variables
```

Globale Variablen werden in den Dateien **`GLOBAL.TDB`** und **`GLOBALS.TDB.SCH`** definiert.

In Mehrbenutzerumgebungen müssen Sie darauf achten, dass alle Clients zur Verwendung derselben im Netzwerk freigegebenen Quelle für globale Variablen oder zur Verwendung von Control Center-basierten globalen Variablen konfiguriert sind.

ANMERKUNG

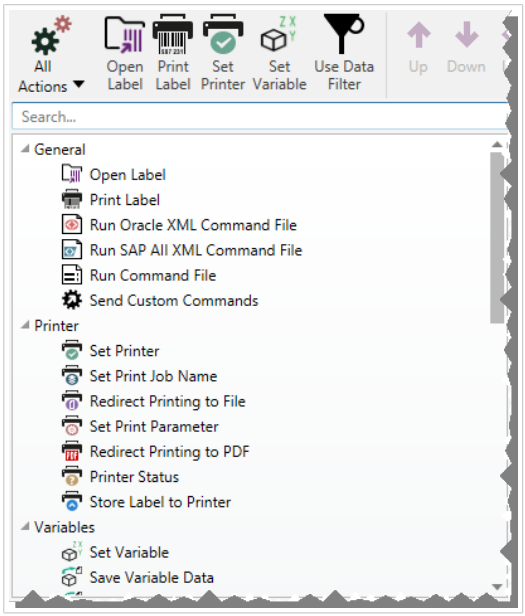
Die Definition und der aktuelle Wert für globale Variablen kann in einer Datei oder im Control Center (für **LMS Enterprise** und **LMS Pro** Produkte) gespeichert werden.

5. Aktionen

Der Abschnitt „Aktionen“ gibt die Liste von Aktionen an, die bei jedem Auslösen des Triggers ausgeführt werden.

5.1. Aktionen definieren

Um eine Aktion zu definieren, klicken Sie auf das Aktions-Symbol in der Gruppe „Aktion einfügen“ in der Multifunktionsleiste. Die Haupt-Multifunktionsleiste enthält häufig verwendete Aktionen. Um alle verfügbaren Aktionen anzuzeigen, klicken Sie auf die Schaltfläche **Alle Aktionen**. Um die verfügbaren Befehle zu der ausgewählten Aktion anzuzeigen, klicken Sie mit der rechten Maustaste auf die Aktion und wählen Sie den gewünschten Befehl aus der Liste.



5.2. Geschachtelte Aktionen

Einige Aktionen können nicht für sich verwendet werden. Ihre spezifischen Funktionen erfordern eine Einbindung unter einer anderen Aktion. Verwenden Sie die Schaltflächen in der Gruppe **Aktionsreihenfolge**, um die Position von Aktionen zu ändern. Jede Aktion wird anhand einer ID-Nummer gekennzeichnet, die ihre Position in der Liste einschließlich Schachtelung anzeigt. Die ID-Nummer wird in der Fehlermeldung angezeigt, sodass Sie die problematische Aktion leichter finden können.

ANMERKUNG

Die Aktion **Etikett drucken** ist ein gutes Beispiel für eine solche Aktion. Sie müssen sie unter der Aktion **Etikett öffnen** positionieren, damit sie das jeweilige zu druckende Etikett referenziert.

5.3. Ausführen von Aktionen

Die Aktionen in der Liste werden pro Trigger einmal ausgeführt. Aufgeführte Aktionen werden von oben nach unten ausgeführt, weswegen die Reihenfolge der Aktionen wichtig ist.

Es gibt zwei Ausnahmen. Die Aktionen **FOR Schleife** und **Datenfilter verwenden** führen geschachtelte Aktionen mehrmals aus. Im Fall von **FOR Schleife** werden Aktionen so häufig ausgeführt, wie in den Aktionseigenschaften festgelegt, und im Fall von **Datenfilter verwenden** einmal für jeden Datensatz in einem vom verbundenen Filter zurückgegebenen Daten-Set.

NiceLabel 10 wird als Dienst unter einem bestimmten Windows-Benutzerkonto ausgeführt und übernimmt die Sicherheitsberechtigungen des Kontos. Weitere Informationen finden Sie im Abschnitt „Im Dienstmodus ausführen“ im NiceLabel Automation Benutzerhandbuch.

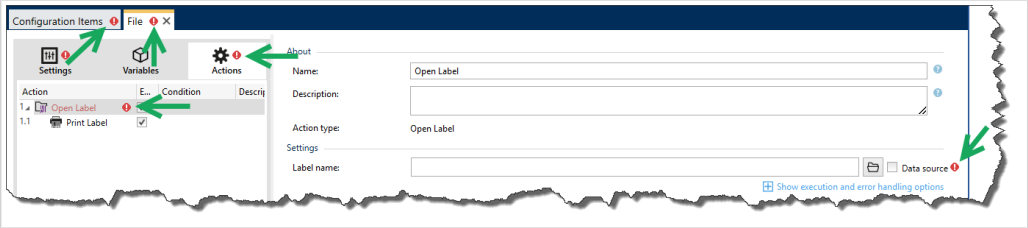
5.4. Bedingungsabhängige Aktionen

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Solche Aktionen werden nur ausgeführt, wenn die festgelegte Bedingung erfüllt ist. Eine Bedingung ist ein einzeiliges Skript (VBScript oder Python). Um Bedingungen festzulegen, klicken Sie in den Aktionseigenschaften auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**, um die Optionen zu erweitern.

In diesem Fall wird die Aktion nur ausgeführt, wenn die vorherige Aktion erfolgreich abgeschlossen wurde, sodass die interne Variable **`ActionLastErrorID`** den Wert 0 aufweist. Um eine solche Bedingung mit internen Variablen zu nutzen, müssen Sie zuerst die interne Variable aktivieren.

5.5. Aktionen mit Konfigurationsfehlern erkennen

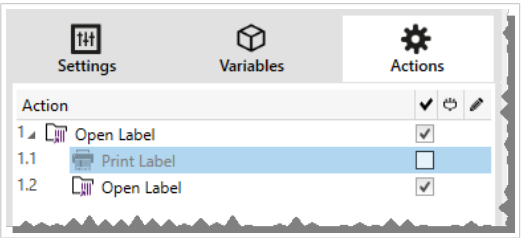
Wenn eine Aktion nicht vollständig konfiguriert ist, wird sie mit einem roten Ausrufezeichen markiert. Solche Aktionen können nicht ausgeführt werden. Sie können solche Aktionen in die **Aktion**-Liste aufnehmen, müssen aber die Konfiguration abschließen, bevor Sie den Trigger starten können. Wenn eine der geschachtelten Aktionen einen Fehlerstatus aufweist, werden auch alle übergeordneten Aktionen (links vom Aktionsnamen) rot markiert, um auf den Fehler in der untergeordneten Aktion hinzuweisen.



In diesem Fall meldet die Aktion **Etikett öffnen** einen Konfigurationsfehler. Es ist kein Parameter für den Etikettennamen angegeben. Das rote Ausrufezeichen wird neben dem fehlerhaften Parameter in der Aktion selbst, in der Liste mit Aktionen, auf der Registerkarte „Aktionen“, auf der Registerkarte „Trigger“ und auf der Registerkarte „Konfigurationselemente“ angezeigt. So können Sie das Problem leicht identifizieren.

5.6. Aktionen deaktivieren

Standardmäßig wird jede neu erstellte Aktion aktiviert und beim Auslösen des Triggers ausgeführt. Sie können Aktionen, die Sie nicht benötigen, deren Konfiguration Sie aber dennoch behalten möchten, deaktivieren. Ein Kürzel für die Aktivierung und Deaktivierung von Aktionen ist das Kontrollkästchen rechts neben dem Aktionsnamen in der Liste definierter Aktionen.



In diesem Fall ist die Aktion **Etikett drucken** noch in der Aktionsliste definiert, wurde aber deaktiviert. Sie wird momentan nicht benötigt und wird bei der Verarbeitung ignoriert, lässt sich aber jederzeit einfach erneut aktivieren.

5.7. Aktionen kopieren

Sie können eine Aktion kopieren und in denselben Trigger oder einen anderen Trigger einfügen. Dazu können Sie die üblichen Windows-Tastenkürzel verwenden oder mit der rechten Maustaste auf die Aktion klicken.

Wenn Sie mit der rechten Maustaste auf die Aktion klicken, werden die verfügbaren Kontextbefehle für das momentan ausgewählte Objekt angezeigt.

Automation Builder ermöglicht es Ihnen außerdem, mehrere Aktionen auszuwählen und die Operationen Kopieren, Einfügen und Löschen auf sie anzuwenden. Um eine Auswahl festzulegen, halten Sie die Strg- und Umschalttaste gedrückt und klicken Sie auf die gewünschten Aktionen.

ANMERKUNG

Mehrere Aktionen können nur unter derselben übergeordneten Aktion ausgewählt werden, d. h. alle ausgewählten Aktionen müssen sich auf derselben Ebene befinden.

5.8. In der Aktionsliste navigieren

Wählen Sie eine definierte Aktion per Mausklick aus und klicken Sie auf die jeweilige Pfeil-Schaltfläche in der Gruppe **Aktionsreihenfolge** in der Multifunktionsleiste. Sie können zur Navigation auch die Tastatur verwenden. Mit den Pfeiltasten verschieben Sie die Auswahl in der Aktionsliste, durch Strg+Pfeiltasten können Sie die Positionen von Aktionen nach oben oder unten bewegen, während die Pfeiltasten nach rechts und links für die Schachtelung genutzt werden.

5.9. Die Aktionen beschreiben

Die Gruppe **Über** ermöglicht Ihnen eine Beschreibung aller NiceLabel 10 Aktionen.

- **Name:** Standardmäßig werden Aktionsnamen nach Typ definiert und sind daher nicht eindeutig. Legen Sie einen benutzerdefinierten Namen fest, um Aktionen in Listen mit anderen Aktionen, in Protokollen und potenziellen Fehlermeldungen leichter unterschieden zu können.
- **Beschreibung:** Benutzerdefinierte Notizen für die ausgewählte Aktion. Die Beschreibung wird im Actions Explorer angezeigt.
- **Aktionstyp:** Nur-Lesen-Feld, das den Typ der Aktion anzeigt.

5.10. Allgemein

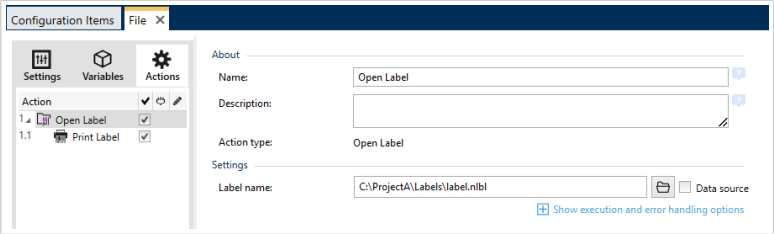
5.10.1. Etikett öffnen

Die Option **Etikett öffnen** gibt die zu druckende Etikettendatei an. Bei Ausführung der Aktion wird die Etikettenvorlage im Zwischenspeicher geöffnet. Das Etikett bleibt im Zwischenspeicher, solange es von Triggern oder Ereignissen verwendet wird.

Es gibt keine Einschränkungen in Bezug auf die Anzahl von Etiketten, die gleichzeitig geöffnet werden können. Falls das Etikett bereits geladen ist und erneut angefordert wird, stellt NiceLabel Automation zuerst fest, ob eine neuere Version verfügbar ist und zum Drucken genehmigt wurde, bevor es das Etikett öffnet.

Im folgenden Beispiel lädt NiceLabel 10 das Etikett **`label.nlbl`** aus dem Ordner **`C:\ProjectA\Labels`**.

```
C:\ProjectA\Labels\label.nlbl
```



Wenn das angegebene Etikett nicht auffindbar ist, versucht NiceLabel 10, es an alternativen Speicherorten zu finden. Weitere Informationen finden Sie im Abschnitt „Suchreihenfolge für angeforderte Dateien“ im NiceLabel Designer Benutzerhandbuch.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Relative Pfade verwenden

NiceLabel 10 unterstützt die Verwendung relativer Pfade zum Referenzieren Ihrer Etikettendatei. Das Stammverzeichnis ist immer der Ordner, in dem die Lösung (bzw. die Konfiguration, falls die Aktion in einer NiceLabel Automation Konfiguration verwendet wird) gespeichert ist.

Mit der folgenden Syntax wird das Etikett relativ zum Speicherort der Konfigurationsdatei geladen. Automation Builder sucht zuerst im Ordner **`ProjectA`** nach dem Etikett, der sich zwei Ebenen über dem aktuellen Ordner befindet, und danach im Ordner **`Labels`**.

```
..\..\ProjectA\Labels\label.nlbl
```

Die Gruppe **Einstellungen** wählt die Etikettendatei aus.

- **Etikettenname:** gibt den Etikettennamen an. Er kann fest codiert werden, woraufhin jedes Mal dasselbe Etikett gedruckt wird. Die Option **Datenquelle** aktiviert die dynamische Definition des Dateinamens. Wählen Sie eine Variable aus (oder fügen Sie eine Variable hinzu), die den Pfad und/oder Dateinamen bereitstellt, wenn ein Trigger ausgeführt wird oder ein Ereignis eintritt.

  TIPP
  Normalerweise wird der Wert der Variablen durch einen Filter zugewiesen.

  ANMERKUNG
  Verwenden Sie für Netzwerkressourcen die UNC-Syntax.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG
  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung

angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.10.2. Etikett drucken

Diese Aktion führt den Etikettendruck aus. Sie muss immer innerhalb der Aktion Etikett öffnen geschachtelt werden. Durch das Schachteln erhält sie die Referenz auf das zu druckende Etikett. So können Sie mehrere Etiketten gleichzeitig geöffnet haben und festlegen, welches Etikett gedruckt werden soll.

Nach Ausführung dieser Aktion wird das Etikett auf dem in der Etikettenvorlage angegebenen Drucker gedruckt. Ist der jeweilige Druckertreiber im System nicht auffindbar, wird das Etikett anhand des Standard-Systemdruckertreibers gedruckt. Sie können die Druckertreiberauswahl anhand der Aktion Drucker einstellen umgehen.

Um Etikettendruck mit hoher Leistung zu erzielen, aktiviert NiceLabel 10 standardmäßig zwei Einstellungen:

- **Parallele Verarbeitung.** Mehrere Druckprozesse werden zeitgleich ausgeführt. Die Anzahl der im Hintergrund ausgeführten Druckprozesse hängt von der Hardware ab, insbesondere vom Prozessortyp. Jeder Prozessorkern kann einen einzelnen Druckthread übernehmen. Diese Standardeinstellung kann geändert werden. Weitere Informationen finden Sie im Abschnitt „Parallele Verarbeitung“ im NiceLabel Automation Benutzerhandbuch.
- **Asynchroner Modus.** Sobald die Trigger-Vorverarbeitung abgeschlossen ist und die Anweisungen für die Druck-Engine verfügbar sind, übernimmt der Druck-Thread die Verarbeitung im Hintergrund. Die Steuerung wird erneut an den Trigger übergeben, damit er den nächsten eingehenden Datenstrom so schnell wie möglich annehmen kann. Ist der synchrone Modus aktiviert, kann der Trigger erst nach Abschluss des Druckprozesses fortfahren. Dies kann einige Zeit dauern, hat aber den Vorteil, dass der Trigger Feedback an die datengebende Anwendung senden kann. Weitere Informationen finden Sie im Abschnitt „Synchroner Modus“ im NiceLabel Automation Benutzerhandbuch.

ANMERKUNG

Die Aktivierung der Option **Fehler in Variable speichern** unter **Aktionsausführung und Fehlerhandhabung** ist im asynchronen Modus wirkungslos, da der Trigger kein Feedback vom Druckprozess erhält. Um Feedback vom Druckprozess zu erfassen, müssen Sie zuerst den synchronen Modus aktivieren.

> **ANMERKUNG**
> Wenn die Aktion „Etikett drucken“ unter einer „FOR Schleife“-Aktion geschachtelt wird, führt Automation sie im Sitzungsdruckmodus aus. Dieser Modus fungiert als Druckoptimierungsmodus, der alle Etiketten anhand einer einzelnen Druckauftragsdatei in einer Schleife druckt. Weitere Informationen finden Sie im Abschnitt „Sitzungsdruck“ des NiceLabel Automation Benutzerhandbuchs.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Menge** gibt die Anzahl von Etiketten an, die anhand der aktiven Eingabemaske gedruckt werden sollen.

- **Etiketten:** legt die Anzahl von gedruckten Etiketten fest. **Datenquelle** legt eine Variable fest (oder erstellt eine Variable), die die Menge für den Etikettendruck dynamisch festlegt.

  > **ANMERKUNG**
  > Der Wert der Variablen wird normalerweise von der Aktion **Datenfilter verwenden** zugewiesen und muss eine Ganzzahl sein.

Alle (unbegrenzte Menge): abhängig vom Etikettenvorlagendesign werden die Etiketten in unterschiedlichen Mengen gedruckt.

Druckdetails für unbegrenzte Menge

Normalerweise wird diese Option in zwei Szenarien eingesetzt.

1. Der Drucker soll kontinuierlich dasselbe Etikett drucken, bis er ausgeschaltet wird oder nachdem er einen Befehl zum Leeren seines Speicherpuffers erhalten hat.

   > **WARNUNG**
   > In diesem Fall muss der NiceLabel Druckertreiber installiert sein und für den Etikettendruck verwendet werden.

   Beim Drucken eines festen Etiketts wird nur ein Druckauftrag an den Drucker gesendet, wobei die Menge auf „endlos“ eingestellt ist. Etikettendrucker haben einen Druckbefehl-Parameter, der „unbegrenztes Drucken“ angibt.

Ist das Etikett nicht fest, sondern enthält Objekte, die sich beim Drucken ändern, z. B. Zähler, wird die gedruckte Menge auf die maximale vom Drucker unterstützte Menge eingestellt. NiceLabelDer kennt die Mengenbeschränkung des Druckers und druckt so viele Etiketten wie möglich.

Beispiel

Die maximal unterstützte Druckmenge beträgt 32.000. Dies ist die Menge von Etiketten, die nach Auswahl der Option Alle (unbegrenzte Menge) gedruckt werden.

2. Der Trigger stellt keine Daten bereit, sondern fungiert nur als Signal, das die Ausführung eines Ereignisses anzeigt. Die Logik zur Anforderung der nötigen Daten befindet sich auf dem Etikett. Normalerweise wird eine Verbindung zu einer Datenbank auf dem Etikett konfiguriert, sodass sich das Etikett bei jeder Auslösung des Triggers mit der Datenbank verbindet und alle Datensätze abruft. In diesem Fall wird die Option **Alle (unbegrenzte Menge)** als Anweisung aufgefasst, alle Datensätze aus der Datenbank zu drucken.

- **Variable Menge (definiert von Etikettenvariable):** gibt eine Etikettenvariable an, die die zu druckende Etikettenmenge bestimmt.
  Der Trigger empfängt die Anzahl zu druckender Etiketten nicht und gibt die Entscheidung daher an die Etikettenvorlage ab. Auf dem Etikett könnte eine Verbindung zu einer Datenbank konfiguriert sein, welche die Etikettenanzahl vorgibt, oder es gibt eine andere Quelle für Mengeninformationen. Eine der Etikettenvariablen muss als „Variable Menge“ definiert sein.

Die Gruppe **Erweitert** definiert die Etikettendruckdetails. Klicken Sie auf **Erweiterte Druckoptionen anzeigen**, um die **Erweiterten** Druckoptionen festzulegen:

In diesem Bereich werden selten genutzte Einstellungen in Bezug auf die Etikettenmenge festgelegt.

- **Anzahl übersprungener Etiketten:** legt die Anzahl von Etiketten fest, die auf der ersten Seite mit Etiketten übersprungen werden sollen. Möglicherweise wurden bereits Etiketten auf den Bogen gedruckt, aber er ist noch nicht vollständig bedruckt. Sie können denselben Bogen erneut verwenden, indem Sie den Versatz für die Startposition angeben. Diese Option ist nützlich, wenn Sie auf Bögen anstelle von Rollen drucken; sie eignet sich also für Bürodrucker, nicht für Etikettendrucker.
- **Identische Etikettenkopien:** legt die Anzahl von Etikettenkopien fest, die für jedes einzelne Etikett gedruckt werden sollen. Bei festen Etiketten führt diese Option zum selben Ergebnis wie die Hauptoption **Etikettenanzahl**. Bei variablen Etiketten, etwa solchen mit Zählern, erhalten Sie echte Etikettenkopien.
- **Etikettensätze:** Gibt vor, wie oft der gesamte Etikettendruckvorgang wiederholt werden soll.

Beispiel

Der Trigger oder das Ereignis empfängt 3 Zeilen CSV-formatierte Daten, wodurch drei Etiketten gedruckt werden (1, 2, 3). Wenn Sie die Option auf 3 einstellen, erfolgt der Ausdruck in der folgenden Reihenfolge: 1, 2, 3, 1, 2, 3, 1, 2, 3.

- **Metadaten**: Mit jedem Druckauftrag werden Ihre Druckkommentare ins Control Center geschrieben. Sie können Ihre Metadaten in der Spalte **Verlauf** > **Drucken** > **Metadaten drucken** sehen. Sie können Metadaten im Control Center zum Sortieren, Filtern und für andere Aufgaben verwenden. Metadaten wirken sich nicht auf Ihren Druck oder Ihre Druckdatenströme aus. Sie können Metadaten zum Protokollieren Ihrer zusätzlichen Informationen über Ihre Druckaufträge im Control Center nutzen. Metadaten können die Losnummer oder andere Labelvariablen, Druckernamen sowie vom Benutzer/System erzeugte Werte umfassen.

  PRODUKTEBENEN-INFO

  Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

TIPP

Alle Erweiterten Gruppenwerte können entweder fest codiert oder durch eine vorhandene oder neu hinzugefügte Variable dynamisch bereitgestellt werden.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.10.3. Oracle XML-Befehlsdatei ausführen

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **LMS Enterprise**.

Diese Aktion führt den Druck mit Daten aus, die aus einer Datei im Oracle XML-Format bezogen wurden.

NiceLabel Automation bietet interne Unterstützung für XML-Dateien mit „Oracle XML“-Struktur, welche in der Oracle Warehouse Management Software definiert werden.

Nutzen Sie diese Aktion als einen Kurzbefehl. Sie hilft Ihnen, Oracle-XML-Dateien direkt auszuführen, ohne sie zuvor anhand eines XML-Filters zu parsen und die Werte Variablen zuzuordnen.

Um diese Aktion nutzen zu können, muss die XML-Datei den Oracle XML-Spezifikationen entsprechen. Weitere Informationen finden Sie im Abschnitt „Oracle XML-Spezifikationen“ im NiceLabel Automation Benutzerhandbuch.

Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch.

PRODUKTEBENEN-INFO

Erfordert **Automation Builder**.

Eine Befehlsdatei in einem Trigger empfangen und ausführen

Wenn ein Trigger eine Befehlsdatei empfangen hat und Sie sie ausführen möchten, führen Sie die folgenden Schritte aus:

1. Klicken Sie im Modul Automation Builder auf der Registerkarte **Variablen** auf die Schaltfläche **Interne Variable** in der Symbolleiste.
2. Aktivieren Sie die interne Variable namens **`DataFileName`** in der Dropdown-Liste. Diese interne Variable stellt den Pfad und den Dateinamen der Datei bereit, welche die vom Trigger

empfangenen Daten enthält. In diesem Fall ist dies die Befehlsdatei. Weitere Informationen finden Sie im Abschnitt Interne Variablen im NiceLabel Automation Benutzerhandbuch.

3. Fügen Sie auf der Registerkarte **Aktionen** die Aktion zur Ausführung der Befehlsdatei hinzu, z. B. Befehlsdatei ausführen, Oracle XML-Befehlsdatei ausführen oder SAP AII XML-Befehlsdatei ausführen.
   Wählen Sie für die Aktion **Befehlsdatei ausführen** den Typ der Befehlsdatei unter **Dateityp** aus.
4. Aktivieren Sie die Option **Variable**.
5. Wählen Sie die Variable namens **`DataFileName`** aus der Liste verfügbarer Variablen.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Datei** definiert die zu verwendende Oracle-XML-Befehlsdatei.

- **Dateiname:** ausgewählte Oracle-XML-Befehlsdatei. Er kann entweder fest codiert oder durch eine vorhandene oder neu erstellte Variable dynamisch definiert werden.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (**`wahr`** oder **`falsch`**). Ist das Ergebnis des Ausdrucks **`wahr`**, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

> **ANMERKUNG**
> Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.10.4. SAP AII XML-Befehlsdatei ausführen

> **PRODUKTEBENEN-INFO**
> Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Diese Aktion führt den Druck mit Daten aus einer Datei im SAP AII XML-Format aus.

NiceLabel Automation bietet interne Unterstützung für XML-Dateien mit der „SAP AII XML“-Struktur, welche in der SAP-Software definiert werden.

Nutzen Sie diese Aktion als einen Kurzbefehl. Sie hilft Ihnen bei der Ausführung von SAP AII XML-Dateien, ohne sie anhand des XML-Filters parsen und die Werte Variablen zuordnen zu müssen. Um diese Aktion nutzen zu können, muss die XML-Datei den SAP AII XML-Spezifikationen entsprechen. Weitere Informationen finden Sie im Abschnitt „SAP AII XML-Spezifikationen“ im NiceLabel Automation Benutzerhandbuch.

Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch.

> **PRODUKTEBENEN-INFO**
> Erfordert **Automation Builder**.

Eine Befehlsdatei in einem Trigger empfangen und ausführen

Wenn ein Trigger eine Befehlsdatei empfangen hat und Sie sie ausführen möchten, führen Sie die folgenden Schritte aus:

1. Klicken Sie im Modul Automation Builder auf der Registerkarte **Variablen** auf die Schaltfläche **Interne Variable** in der Symbolleiste.
2. Aktivieren Sie die interne Variable namens **`DataFileName`** in der Dropdown-Liste. Diese interne Variable stellt den Pfad und den Dateinamen der Datei bereit, welche die vom Trigger empfangenen Daten enthält. In diesem Fall ist dies die Befehlsdatei. Weitere Informationen finden Sie im Abschnitt Interne Variablen im NiceLabel Automation Benutzerhandbuch.
3. Fügen Sie auf der Registerkarte **Aktionen** die Aktion zur Ausführung der Befehlsdatei hinzu, z. B. Befehlsdatei ausführen, Oracle XML-Befehlsdatei ausführen oder SAP AII XML-Befehlsdatei ausführen.
   Wählen Sie für die Aktion **Befehlsdatei ausführen** den Typ der Befehlsdatei unter **Dateityp** aus.
4. Aktivieren Sie die Option **Variable**.
5. Wählen Sie die Variable namens **`DataFileName`** aus der Liste verfügbarer Variablen.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Datei** definiert die zu verwendende SAP AII XML-Befehlsdatei.

- **Dateiname:** ausgewählte SAP AII XML-Befehlsdatei. Er kann entweder fest codiert oder durch eine vorhandene oder neu erstellte Variable dynamisch definiert werden.

Die Gruppe **Optionale Parameter** ermöglicht es Ihnen, den Etikettennamen festzulegen, sofern dieser nicht in der XML-Datei enthalten ist.

- **Etikettenname:** die ausgewählte Etikettendatei, die in der Befehlsdatei verwendet werden soll. Er kann entweder fest codiert oder durch eine vorhandene oder neu erstellte Variable dynamisch definiert werden.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.10.5. Befehlsdatei ausführen

Diese Aktion führt Befehle aus, die in einer ausgewählten Befehlsdatei enthalten sind. **Alle Dateityp**-Optionen stellen Befehle bereit, die NiceLabel 10 von oben nach unten ausführt.

Befehlsdateien enthalten für gewöhnlich Daten für ein einziges Etikett, aber Sie können Dateien mit beliebiger Komplexität erstellen. Weitere Informationen finden Sie im Abschnitt „Typen von Befehlsdateien“.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.

- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die **Datei**-Gruppe legt den Typ und den Namen der Befehlsdatei fest, die ausgeführt werden soll (JOB, XML oder CSV).

- **Dateityp.** Gibt den Typ der auszuführenden Befehlsdatei an.
- **Dateiname.** Gibt den Namen der Befehlsdatei an.
  Der **Dateiname** kann fest codiert werden, woraufhin jedes Mal dieselbe Befehlsdatei ausgeführt wird. Die Option **Variable** aktiviert einen variablen Dateinamen. Wählen Sie eine Variable aus (oder erstellen Sie eine Variable), die den Pfad und/oder Dateinamen bereitstellt, wenn ein Trigger ausgeführt wird oder ein Ereignis eintritt. Normalerweise wird der Wert der Variablen durch einen Filter zugewiesen.
  Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch.

PRODUKTEBENEN-INFO
Erfordert **Automation Builder**.

Eine Befehlsdatei in einem Trigger empfangen und ausführen

Wenn ein Trigger eine Befehlsdatei empfangen hat und Sie sie ausführen möchten, führen Sie die folgenden Schritte aus:

1. Klicken Sie im Modul Automation Builder auf der Registerkarte **Variablen** auf die Schaltfläche **Interne Variable** in der Symbolleiste.
2. Aktivieren Sie die interne Variable namens **`DataFileName`** in der Dropdown-Liste. Diese interne Variable stellt den Pfad und den Dateinamen der Datei bereit, welche die vom Trigger empfangenen Daten enthält. In diesem Fall ist dies die Befehlsdatei. Weitere Informationen finden Sie im Abschnitt Interne Variablen im NiceLabel Automation Benutzerhandbuch.
3. Fügen Sie auf der Registerkarte **Aktionen** die Aktion zur Ausführung der Befehlsdatei hinzu, z. B. Befehlsdatei ausführen, Oracle XML-Befehlsdatei ausführen oder SAP AII XML-Befehlsdatei ausführen.
   Wählen Sie für die Aktion **Befehlsdatei ausführen** den Typ der Befehlsdatei unter **Dateityp** aus.
4. Aktivieren Sie die Option **Variable**.
5. Wählen Sie die Variable namens **`DataFileName`** aus der Liste verfügbarer Variablen.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.10.6. Benutzerdefinierte Befehle senden

Diese Aktion führt die eingegebenen benutzerdefinierten NiceLabel Befehle aus.

Schachteln Sie diese Aktion immer unter der Aktion Etikett öffnen. Dies ermöglicht eine Referenzierung des Etiketts, auf das die Befehle angewandt werden. Weitere Informationen finden Sie im Abschnitt „Benutzerdefinierte Befehle verwenden“ des NiceLabel Automation Benutzerhandbuchs.

ANMERKUNG

Die meisten benutzerdefinierten Befehle sind in einzelnen Aktionen verfügbar, sodass Sie in den meisten Fällen keine benutzerdefinierten Befehle benötigen.

ANMERKUNG

Die Aktion „Benutzerdefinierte Befehle senden“ kann verwendet werden, um den Sitzungsdruckmodus zu beenden. Dieser Modus fungiert als Druckoptimierungsmodus, der alle Etiketten anhand einer einzelnen Druckauftragsdatei in einer Schleife druckt. Um den Sitzungsdruck zu beenden, schachteln Sie die Aktion „Benutzerdefinierte Befehle senden“ unter den Aktion „FOR Schleife“ und verwenden Sie den Befehl SESSIONEND. Weitere Informationen finden Sie in den Abschnitten „Sitzungsdruck“ und „Benutzerdefinierte Befehle verwenden“ im NiceLabel Automation Benutzerhandbuch.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Der **Skript**-Editor bietet die folgenden Funktionen:

- **Datenquelle einfügen**: fügt eine vorhandene oder neu erstellte Variable in das Skript ein.
- **Skript-Editor:** öffnet den Editor, der Scripting einfacher und effizienter macht.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.11. Drucker

5.11.1. Drucker einstellen

Diese Aktion gibt den Namen des Druckers an, der zum Drucken des aktiven Etiketts verwendet werden soll.

> **ANMERKUNG**
> Diese Aktion übergeht den in den Etiketteneigenschaften ausgewählten Drucker.

Diese Aktion ist nützlich wenn dasselbe Etikett auf mehreren Druckern gedruckt werden soll. Schachteln Sie diese Aktion immer unter der Aktion Etikett öffnen, um dem Etikett die Referenz auf den bevorzugten Drucker mitzuteilen.

Diese Aktion liest die Standardeinstellungen (wie Druckgeschwindigkeit und Temperatur) aus dem ausgewählten Druckertreiber und wendet sie auf das Etikett an. Wenn Sie die Aktion **Drucker einstellen** nicht nutzen, wird das Etikett auf dem Drucker gedruckt, der in der Etikettenvorlage definiert ist.

WARNUNG

Seien Sie beim Wechseln der Drucker vorsichtig, etwa bei einer Änderung von Zebra auf SATO, und auch bei Änderungen zwischen verschiedenen Druckermodellen derselben Marke. Die Druckereinstellungen sind eventuell nicht kompatibel und die Ausdrucke der Etiketten nicht identisch. Außerdem stehen Etikettendesign-Optimierungen wie interne Zähler und interne Schriften unter Umständen auf dem neu ausgewählten Drucker nicht zur Verfügung.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Drucker** gibt den Namen des für den aktuellen Druckauftrag zu verwendenden Druckers an.

- **Druckername**: Wählen Sie ihn aus der Liste lokal installierter Druckertreiber aus oder geben Sie einen Druckernamen ein. Wählen Sie **Datenquelle** aus, um den Drucker anhand einer Variablen dynamisch auszuwählen. Ist diese Option aktiviert, wählen Sie eine Variable aus (oder erstellen Sie eine Variable), die den bei Ausführung der Aktion genutzten Druckernamen enthält.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

> **ANMERKUNG**
> Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.11.2. Druckauftragsnamen festlegen

Diese Aktion gibt den Namen der Druckauftragsdatei an, wie sie im Windows Spooler erscheint. Der Standard-Druckauftragsname ist der Name der verwendeten Etikettendatei. Er wird von dieser Aktion überschrieben.

> **ANMERKUNG**
> Sie müssen diese Aktion immer unter der Aktion Etikett öffnen einbinden, damit sie auf die richtige Etikettendatei angewandt wird.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Druckauftrag** gibt den Namen des Druckauftrags an.

- **Name:** legt den Namen des Druckauftrags fest. Er kann fest codiert werden, woraufhin derselbe Name für jede Druckaktion verwendet wird. „Variable“ aktiviert einen variablen Dateinamen. Wählen

Sie eine Variable aus (oder erstellen Sie eine Variable), die den Pfad und/oder Dateinamen bereitstellt, wenn ein Trigger ausgeführt wird oder ein Ereignis eintritt.

ANMERKUNG

Im Modul Automation Builder wird der Variablenwert normalerweise durch einen Filter zugewiesen.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.11.3. Druck an Datei umleiten

Diese Aktion leitet den Druckauftrag an eine Datei weiter. Die erstellte Druckdatei wird nicht an eine im Druckertreiber definierte Druckerschnittstelle gesendet, sondern an eine Datei umgeleitet. Sie können Daten an den Inhalt einer vorhandenen Datei anhängen oder diese überschreiben.

Mithilfe dieser Aktion können Sie Druckerbefehle in einer separaten Datei erfassen.

Die Aktion weist das Modul Automation Builder an, den Druck umzuleiten – dies führt dazu, dass die Etiketten nicht gedruckt werden. Stellen Sie sicher, dass auf diese Aktion die Aktion Etikett drucken folgt.

ANMERKUNG

NiceLabel Automation wird unter einem bestimmten Windows-Benutzerkonto als Dienst ausgeführt. Stellen Sie sicher, dass dieses Konto über Lese-/Schreibzugriff auf den jeweiligen Ordner verfügt. Weitere Informationen finden Sie im Abschnitt „Zugriff auf freigegebene Ressourcen im Netzwerk“ im NiceLabel Automation Benutzerhandbuch.

ANMERKUNG

Die Aktion Druck an Datei umleiten ist nützlich, um mehrere verschiedene Etiketten (.NLBL-Dateien) auf einem Netzwerkdrucker zu drucken und dabei die korrekte Reihenfolge der Etiketten aufrechtzuerhalten. Wenn mehrere .NLBL-Dateien vom selben Trigger gedruckt werden, sendet Automation Builder jedes Etikett in einem separaten Druckauftrag an den Drucker, selbst wenn derselbe Zieldrucker für die Etiketten verwendet wird. Wenn ein Netzwerkdrucker verwendet wird, kann ein Auftrag eines anderen Benutzers zwischen zwei anderen Aufträgen eingefügt werden. Anhand dieser Aktion können Sie Druckdaten in dieselbe Datei einfügen und deren Inhalt dann anhand der Aktion Daten an Drucker senden an den Drucker senden.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Einstellungen-Gruppe **Datei** legt die Art und Weise der Dateiauswahl für die Umleitung fest.

- **Dateiname:** gibt den Dateinamen an. Er kann entweder fest codiert oder durch eine vorhandene oder neu erstellte Variable dynamisch definiert werden.
  Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch.

> **ANMERKUNG**
> Stellen Sie bei Verwendung dieser Aktion sicher, dass die Berechtigungen Ihres Benutzerkonto Ihnen den Lese-/Schreibzugriff auf den jeweiligen Ordner ermöglichen.

Die Einstellungen-Gruppe **Datei-Schreibmodus** wählt aus, wie die Datei im Fall von wiederholten Umleitungen behandelt wird.

- **Datei überschreiben**: Falls die angegebene Datei bereits auf der Festplatte vorhanden ist, wird sie überschrieben.
- **Füge Daten an Datei**: die Auftragsdatei wird den vorhandenen Daten in der jeweiligen Datei hinzugefügt.

Die Gruppe **Dauerhaftigkeit** steuert die Kontinuität der Umleitungsaktion. Dieser Parameter definiert die Anzahl von Etikett drucken-Aktionen, auf die sich die Aktion **Druck an Datei umleiten** auswirkt.

- **Auf nächste Druckaktion anwenden**: legt fest, dass die Druckumleitung nur auf die nächste Etikett drucken-Aktion angewandt wird (einzelnes Ereignis).
- **Auf alle folgenden Druckaktionen anwenden**: legt fest, dass die Druckumleitung auf alle **Etikett drucken**-Aktionen angewandt wird, die nach der aktuellen **Druck an Datei umleiten**-Aktion definiert sind.

> **ANMERKUNG**
> Diese Aktion leitet den Druck lediglich um. Stellen Sie sicher, dass darauf die Aktion Etikett drucken folgt.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.11.4. Druckparameter festlegen

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Mit dieser Aktion können Sie eine Feinabstimmung der Parameter in Bezug auf den Druckertreiber vornehmen. Dazu zählen Parameter wie Geschwindigkeit und Temperatur für Etikettendrucker oder Papierschacht für Laserdrucker.

Die Druckereinstellungen werden nur auf den aktuellen Druckvorgang angewandt und beim nächsten Ereignis nicht berücksichtigt.

WARNUNG

Ihre Parameter aus der Aktion **Druckparameter festlegen** werden aus Ihrem Control Center bei einer Vorschau oder erneutem Etikettendruck nicht übernommen.

Sie können dies vermeiden, indem Sie Ihre Druckparameter (Druckereigenschaften) in Ihrer Etikettenvorlage oder Ihrem Druckertreiber speichern. Die Druckereigenschaften können sich von Treiber zu Treiber und auch von Drucker zu Drucker unterscheiden.

ANMERKUNG

Wenn Sie die Aktion Druckparameter festlegen nutzen, um den Druckernamen zu ändern, stellen Sie sicher, dass die Aktion **Druckparameter festlegen** gleich im Anschluss verwendet wird. Bevor Sie die DEVMODE-Struktur auf den Druckertreiber anwenden können, müssen Sie zuerst die Standard-Treibereinstellungen laden. Dies erfolgt anhand der Aktion „Drucker einstellen“. Der DEVMODE ist nur mit dem DEVMODE desselben Druckertreibers kompatibel.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Druckparameter** ermöglicht eine Feinanpassung der Aktion vor dem Druck.

- **Papierfach:** Name des Papierfachs, in dem sich die Etikettenmedien befinden. Diese Option wird normalerweise für Laser- und Tintenstrahldrucker mit mehreren Papierfächern verwendet. Der angegebene Name des Papierfachs muss dem Namen des Fachs im Druckertreiber entsprechen. In den Druckertreiber-Eigenschaften finden Sie weitere Details.
- **Druckgeschwindigkeit**: legt die Druckgeschwindigkeit fest. Diese Einstellung umgeht die im Etikett definierte Einstellung. Der angegebene Wert muss im Bereich der akzeptierten Werte liegen.

Beispiel

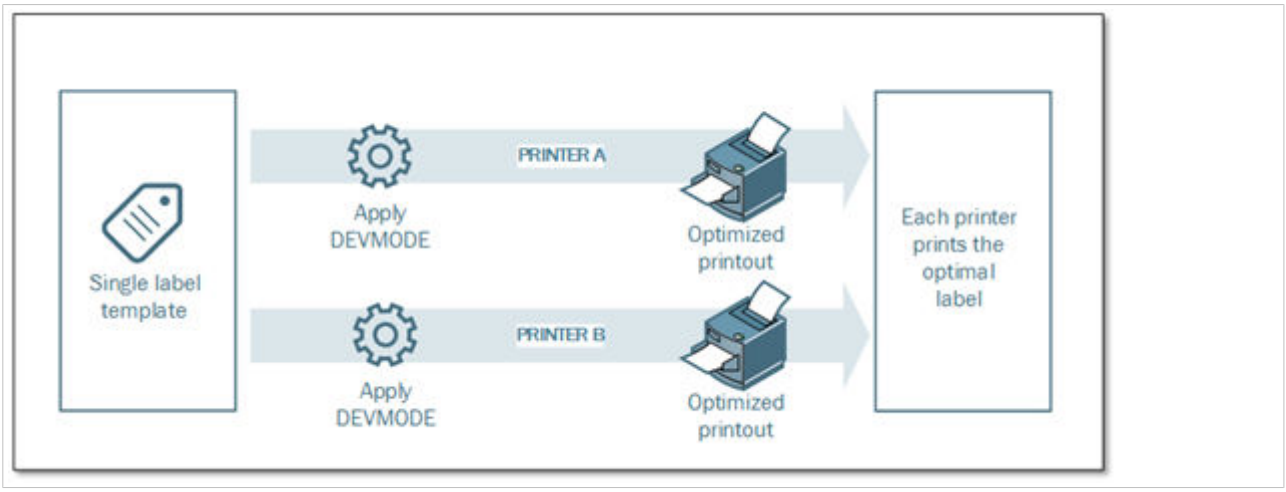
Das erste Druckermodell akzeptiert Werte zwischen 0 und 30, während das zweite Werte zwischen -15 und 15 akzeptiert. Weitere Informationen finden Sie unter Druckertreiber-Einstellungen.

- **Temperatur**: definiert die Temperatur der gedruckten Objekte auf dem Papier und umgeht die Einstellungen vom Etikett. Der angegebene Wert muss im Bereich der akzeptierten Werte liegen.
- **Druckversatz X**: gibt den horizontalen Versatz vor. Der Etikettendruck wird um die angegebene Anzahl von Punkten in horizontaler Richtung versetzt. Es kann ein negativer Versatz festgelegt werden.
- **Druckversatz Y**: gibt den vertikalen Versatz vor. Der Etikettendruck wird um die angegebene Anzahl von Punkten in vertikaler Richtung versetzt. Es kann ein negativer Versatz festgelegt werden.

TIPP

Alle Druckparameter können entweder fest codiert oder durch eine vorhandene oder neu erstellte Variable dynamisch definiert werden.

In der Gruppe **Erweitert** werden die mit dem Druckauftrag gesendeten Druckereinstellungen angepasst.



Druckereinstellungen wie Druckgeschwindigkeit, Temperatur, Medientyp, Versätze usw. können wie folgt definiert werden:

- In einem Etikett
- Per Abruf aus einem Druckertreiber
- Per Abruf von einem Drucker zum Zeitpunkt des Drucks

Die unterstützten Methoden hängen von den Möglichkeiten des Druckertreibers ab. Der Druckmodus (Einstellungen vom Etikett oder Treiber oder Drucker abrufen) kann im Etikettendesign konfiguriert werden. Eventuell müssen Sie diese Druckereinstellungen zum Zeitpunkt des Drucks anwenden – sie können für jeden Druckvorgang abweichen.

Beispiel

Ein einzelnes Etikett sollte auf verschiedenen Druckern gedruckt werden, aber jeder Drucker erfordert geringfügig andere Parameter. Drucker unterschiedlicher Hersteller nutzen nicht dieselben Werte zur Einstellung der Druckgeschwindigkeit oder der Temperatur. Zudem erfordern einige Drucker einen vertikalen oder horizontalen Versatz, um das Etikett an der richtigen Position zu drucken. Während der Testphase können Sie die optimalen Einstellungen für jeden Drucker festlegen, den Sie nutzen möchten, und sie direkt vor dem Drucken auf eine einzige Etikettenvorlage anwenden. Diese Aktion wendet die entsprechenden Einstellungen auf jeden definierten Drucker an.

Diese Aktion rechnet damit, die Druckereinstellungen in einer DEVMODE-Struktur zu empfangen. Dabei handelt es sich um eine Windows-Standard-Datenstruktur mit Informationen zur Initialisierung und Umgebung eines Druckers.

Die Option **Druckereinstellungen** wendet benutzerdefinierte Druckereinstellungen an. Die folgenden Eingabemethoden sind verfügbar:

- **Base64-codierte Festdaten mit DEVMODE-Struktur**. In diesem Fall geben Sie den DEVMODE des Druckers als Base64-codierte Zeichenfolge direkt in das Bearbeitungsfeld ein. Bei der Ausführung konvertiert die Aktion die Base64-codierten Daten zurück ins Binärformat.

- **Base64-codierte variable Daten mit DEVMODE-Struktur**. In diesem Fall muss die ausgewählte Datenquelle die Base64-codierte DEVMODE-Struktur enthalten. Aktivieren Sie Datenquelle und wählen Sie die jeweilige Variable aus der Liste aus. Bei der Ausführung konvertiert die Aktion die Base64-codierten Daten zurück ins Binärformat.
- **Binäre variable Daten mit DEVMODE-Struktur (verfügbar in Automation Builder)**. In diesem Fall muss die ausgewählte Variable die DEVMODE-Struktur im nativen Binärformat enthalten. Aktivieren Sie **Datenquelle** und wählen Sie die jeweilige Variable aus der Liste aus. Bei der Ausführung nutzt die Aktion die empfangene DEVMODE-Struktur, ohne dass eine Konvertierung stattfindet.

  ANMERKUNG

  Wenn die ausgewählte Variable keine binäre DEVMODE-Struktur bereitstellt, müssen Sie sicherstellen, dass sie in der Konfiguration als binäre Variable definiert ist.

ANMERKUNG

Stellen Sie sicher, dass vor dieser Aktion die Aktion Drucker einstellen definiert ist.

Label Settings übersteuert die in **Label Properties** in Designer definierten Etiketten-Eigenschaften. Verwenden Sie diese Option, um Ihre Etiketten an einem Drucker oder Datenträger mit anderen als den in **Label Properties** in Designer definierten Eigenschaften auszudrucken. Mit dieser Option haben Sie folgende Möglichkeiten:

- Etikettenmaße (Breite und Höhe) ändern.
- Etiketten-Seitenrand hinzufügen oder ändern.
- Cutter deaktivieren.
- Stapeldruck deaktivieren.
- Verschiedene Bestände anwenden, indem Sie den Parameter **Labels Across** ändern (horizontale und vertikale Anzahl, Lücken, Verarbeitungsreihenfolge).
- Hochformat bzw. Querformat neu definieren.
- Etiketten um 180° drehen.

Die Automatisierung wendet die **Label Settings** zum Druckzeitpunkt an. Die Parameter der Etiketten-Einstellungen werden in Ihren Etiketten-Vorlagen nicht abgespeichert. Sie können Etiketten-Einstellungen als XML Nutzlast bereitstellen.

Beispiel für Etiketten-Einstellungen XML

Das folgende Beispiel zeigt eine Strukturansicht der Etiketteneinstellungen und ihrer Attribute.

> **ANMERKUNG**
> Die Maßeinheiten in XML entsprechen Ihren Etikettendesign-Einheiten (cm, in, mm, Punkt). Sie können die Einheiten Designer ändern, indem Sie auf **Label Properties** > **Label Dimensions** > **Unit of measure** gehen.

```
<LabelSettings>
    <Width>100</Width>
    <Height>30</Height>
    <Margin>
        <Left>2</Left>
        <Right>3</Right>
        <Top>4</Top>
        <Bottom>5</Bottom>
    </Margin>
    <LabelsAcross>
        <Horizontal>
            <Count>2</Count>
            <Gap>4</Gap>
        </Horizontal>
        <Vertical>
            <Count>3</Count>
            <Gap>5</Gap>
        </Vertical>
        <ProcessingOrder>HorizontalTopRight</ProcessingOrder>
    </LabelsAcross>
    <Orientation>Landscape</Orientation>
    <Rotated>true</Rotated>
    <DisableCutter/>
    <DisableBatchPrinting/>
</LabelSettings>
```

Etiketten-Einstellungen XML Spezifikation

Dieser Abschnitt enthält eine Beschreibung der XML Dateistruktur, um die Parameter und Werte von **Label Settings** zu definieren.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="nonNegativeFloat">
    <xs:restriction base="xs:float">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="LabelSettings">
    <xs:complexType>
```

```
      <xs:all>
        <xs:element name="DisableCutter" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence/>
          </xs:complexType>
        </xs:element>
        <xs:element name="DisableBatchPrinting" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence/>
          </xs:complexType>
        </xs:element>
        <xs:element name="Width" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Height" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Margin" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element name="Left" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
              <xs:element name="Right"  type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
              <xs:element name="Top"  type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
              <xs:element name="Bottom" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element name="LabelsAcross" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element name="Horizontal" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="Count" type="xs:nonNegativeInteger"
minOccurs="0" maxOccurs="1" />
                      <xs:element name="Gap" type="nonNegativeFloat"
minOccurs="0" maxOccurs="1"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
              <xs:element name="Vertical" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="Count" type="xs:nonNegativeInteger"
```

```
minOccurs="0" maxOccurs="1" />
                    <xs:element name="Gap" type="nonNegativeFloat"
minOccurs="0" maxOccurs="1"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
              <xs:element name="ProcessingOrder" minOccurs="0" maxOccurs="1">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="HorizontalTopLeft"/>
                    <xs:enumeration value="HorizontalTopRight"/>
                    <xs:enumeration value="HorizontalBottomLeft"/>
                    <xs:enumeration value="HorizontalBottomRight"/>
                    <xs:enumeration value="VerticalTopLeft"/>
                    <xs:enumeration value="VerticalTopRight"/>
                    <xs:enumeration value="VerticalBottomLeft"/>
                    <xs:enumeration value="VerticalBottomRight"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element name="Orientation" minOccurs="0" maxOccurs="1">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Portrait"/>
              <xs:enumeration value="Landscape"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Rotated" type="xs:boolean" minOccurs="0"
maxOccurs="1"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.

- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.11.5. Druckumleitung an PDF

PRODUKTEBENEN-INFO

Diese Aktion ist in NiceLabelLMS Enterprise verfügbar.

Diese Aktion leitet den Druckauftrag an ein PDF-Dokument weiter. Das so erstellte PDF-Dokument weist exakt dieselben Etiketten-Abmessungen auf, die bei der Erstellung des Etiketts vorgegeben wurden. Die Render-Qualität von Grafiken im PDF entspricht der Auflösung des Zieldruckers und der gewünschten Druckgröße.

Druckstromdaten können zu einer vorhandenen Datei hinzugefügt werden oder sie überschreiben.

Die Aktion weist NiceLabel 10 an, den Druck umzuleiten – dies führt dazu, dass die Etiketten nicht gedruckt werden. Stellen Sie sicher, dass auf diese Aktion die Aktion **Etikett drucken** folgt.

> **ANMERKUNG**
> Das Modul NiceLabel Automation wird unter dem definierten Windows-Benutzerkonto als Dienst ausgeführt. Stellen Sie sicher, dass dieses Konto über Lese-/Schreibzugriff auf den jeweiligen Ordner verfügt. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Datei** legt die Datei fest, an die umgeleitet wird.

- **Dateiname**: gibt den Namen der Datei an, an die der Druckauftrag umgeleitet wird. Ist er fest codiert, die der Druckstrom jedes Mal an dieselbe angegebene Datei geleitet. Um ihn dynamisch festzulegen, verwenden Sie eine vorhandene Variable oder erstellen Sie eine neue.
- **Datei überschreiben**: Falls die angegebene Datei bereits auf der Festplatte vorhanden ist, wird sie überschrieben (standardmäßig ausgewählt).
- **Füge Daten an Datei**: Die Auftragsdatei wird den vorhandenen Daten in der jeweiligen Datei hinzugefügt (standardmäßig nicht ausgewählt).
- **Schriftarten in PDF einbetten**: Wenn Sie eine nicht standardisierte Schriftart verwenden, kann Ihre Lösung auf Computern ohne diese Schriftart eine andere PDF-Ausgabe erzeugen. Wenn die Option **Schriftarten in PDF einbetten** aktiviert ist, betten Sie Ihre nicht standardisierte Schriftart in Ihre Lösung ein, welche daraufhin auf allen Computern die gleiche PDF-Datei erstellt.
  Die Option **Schriftarten in PDF einbetten** erstellt ein PDF für die Archivierung (PDF/A) konformes Dokument. Der PDF/A-Standard stellt sicher, dass Ihre Dokumente unabhängig von der verwendeten Software auf genau dieselbe Weise reproduziert werden. Die für die Darstellung der Inhalte erforderlichen Informationen wie Bilder, Schriftarten und Farbinformationen sind in Ihrer PDF-Datei eingebettet.

  > **ANMERKUNG**
  > Die PDF-Ausgabedatei verbraucht mehr Speicherplatz, wenn sie mit dieser Option gespeichert wird.

Die Gruppe **Dauerhaftigkeit** steuert die Dauerhaftigkeit der Umleitungsaktion. Dieser Parameter definiert die Anzahl von Etikett drucken-Aktionen, auf die sich die Aktion **Druck an Datei umleiten** auswirkt.

- **Auf nächste Druckaktion anwenden**: legt fest, dass die Druckumleitung nur auf die nächste Etikett drucken-Aktion angewandt wird (einzelnes Ereignis).
- **Auf alle folgenden Druckaktionen anwenden**: legt fest, dass die Druckumleitung auf alle Etikett drucken-Aktionen angewandt wird, die nach der aktuellen Druck an Datei umleiten-Aktion definiert sind.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.11.6. Druckerstatus

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Diese Aktion kommuniziert mit dem Drucker, um dessen Echtzeit-Status zu erhalten, und kontaktiert den Windows Spooler, um weitere Informationen über den Drucker und dessen Aufträge abzurufen.

So werden Informationen zu Fehlern, zum Spooler-Status und der Anzahl von Aufträgen im Spooler erhoben.. Auf diese Weise lassen sich potenzielle Fehler leicht auffinden.

ANMERKUNG

Beispiele für Verwendungsmöglichkeiten. (1) Druckerstatus vor dem Drucken überprüfen. Falls der Drucker einen Fehlerstatus aufweist, wird das Etikett auf dem Backup-Drucker gedruckt. (2) Aufträge im Spooler des Hauptdruckers zählen. Falls sich zu viele darin befinden, drucken Sie das Etikett auf einem alternativen Drucker. (3) Sie verifizieren den Druckerstatus vor dem Drucken. Falls der Drucker einen Fehlerstatus aufweist, drucken Sie keine Etiketten, sondern melden den Fehler anhand einer Übermittlungsaktion an das Hauptsystem, z. B. über Daten an TCP/IP-Port senden, Daten an HTTP senden, SQL-Anweisung ausführen, Webdienst, oder aber innerhalb der Trigger-Antwort.

Voraussetzungen für Echtzeit-Druckerstatus

Um eine Überwachung des Druckerstatus in Echtzeit zu ermöglichen, folgen Sie diesen Anweisungen:

- Verwenden Sie den NiceLabel Druckertreiber, um detaillierte Statusinformationen zu erhalten. Wenn Sie einen anderen Druckertreiber verwenden, können Sie nur die vom Windows Spooler abgerufenen Parameter überwachen.
- Der Drucker muss in der Lage sein, seinen Echtzeitstatus zu melden. Für Druckermodelle mit Unterstützung für bidirektionale Kommunikation, siehe NiceLabel Download-Webseite.
- Der Drucker muss mit einer Schnittstelle verbunden sein, die bidirektionale Kommunikation unterstützt.
- Die bidirektionale Unterstützung muss unter **Systemsteuerung > Hardware und Sound > Geräte und Drucker anzeigen > Treiber > Druckereigenschaften > Anschlüsse-Registerkarte > Bidirektionale Unterstützung aktivieren** aktiviert werden.
- Wenn Sie einen mit dem Netzwerk verbundenen Etikettendrucker verwenden, stellen Sie sicher, dass Sie **Advanced TCP/IP Port** verwenden, nicht **Standard TCP/IP Port**. Weitere Informationen finden Sie im Knowledge Base-Artikel.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Drucker** wählt den Drucker aus.

- **Druckername:** gibt den Namen des für den aktuellen Druckauftrag zu verwendenden Druckers an. Sie können einen Drucker aus der Liste lokal installierter Druckertreiber auswählen oder einen Druckernamen eingeben. „Datenquelle“ aktiviert den variablen Dateinamen. Wählen Sie nach Aktivierung eine Variable aus (oder erstellen Sie eine Variable), die den Druckernamen bereitstellt, wenn ein Trigger ausgeführt wird oder ein Ereignis eintritt. Normalerweise wird der Variablenwert durch einen Filter zugewiesen.

Die Gruppe **Datenzuordnung** legt die Parameter fest, die als Ergebnis der Aktion **Druckerstatus** ausgegeben werden.

WARNUNG

Die meisten der folgenden Parameter werden nur mit dem NiceLabel Druckertreiber unterstützt. Wenn Sie einen anderen Druckertreiber nutzen, können Sie nur die Spooler-bezogenen Parameter verwenden.

- **Druckerstatus** gibt den Echtzeitstatus des Druckers als Zeichenfolge formatiert an.
  Gibt der Drucker mehrere Status aus, werden alle davon, durch Kommas („,“) getrennt, zu einer Zeichenfolge verbunden. Werden keine Druckerstatus gemeldet, bleibt dieses Feld leer. Mögliche Druckerstatus sind **Offline**, **Keine Etiketten** oder **Farbband fast verbraucht**. Da es kein standardisiertes Meldungsprotokoll gibt, nutzt jeder Druckeranbieter eigene Statusmeldungen.
- **Druckerfehler:** boolescher Wert (wahr/falsch) des „Druckerfehler“-Status.
- **Drucker offline:** boolescher Wert (wahr/falsch) des „Drucker offline“-Status.
- **Treiber angehalten:** boolescher Wert (wahr/falsch) des „Treiber angehalten“-Status.
- **NiceLabel Treiber:** boolescher Wert (wahr/falsch) des „ Treiber“-Status. Gibt an, ob es sich beim ausgewählten Treiber um einen NiceLabel Treiber handelt.
- **Spoolerstatus:** gibt den Spoolerstatus in Form einer Zeichenfolge gemäß der Meldung des Windows-Systems an. Der Spooler kann mehrere Status gleichzeitig melden. In diesem Fall werden die Status durch Kommas „,“) getrennt.
- **Spoolerstatus-ID:** gibt den Spoolerstatus in Form einer Zahl gemäß der Meldung des Windows-Systems an. Der Spooler kann mehrere Status gleichzeitig melden. In diesem Fall enthält die ausgegebene Statusmeldung alle zutreffenden IDs als Zahlenwerte. Der Wert 5 zum Beispiel steht für die Status-IDs 4 und 1, d. h.: Drucker weist einen Fehler auf, Drucker ist angehalten. Siehe folgende Tabelle.

TIPP

Die Aktion gibt dezimale Werte aus, während die Werte in der folgenden Tabelle hexadezimal sind; sie müssen sie also konvertieren, bevor Sie die Antwort parsen.

- **Tabelle mit Spoolerstatus-IDs und ihren Beschreibungen**

Spoolerstatus-ID (als Hexadezimalwert)	Spoolerstatus-Beschreibung
0	Kein Status.
1	Drucker ist angehalten.
2	Drucker druckt.
4	Drucker weist einen Fehler auf.
8	Drucker ist nicht verfügbar.
10	Drucker hat kein Papier mehr.
20	Manuelle Zufuhr erforderlich.
40	Drucker hat ein Papierproblem.
80	Drucker ist offline.
100	Aktiver Eingabe-/Ausgabestatus.
200	Drucker ist beschäftigt.
400	Papierstau.
800	Ausgabeschacht ist voll.
2000	Drucker wartet.
4000	Drucker verarbeitet Daten.
10000	Drucker fährt hoch.
20000	Toner-/Tintenstand ist niedrig.
40000	Kein Toner im Drucker.
80000	Aktuelle Seite kann nicht gedruckt werden.
100000	Benutzereingriff erforderlich.
200000	Druckerspeicher voll.
400000	Klappe geöffnet.
800000	Unbekannter Fehler.
1000000	Drucker ist im Energiesparmodus.

- **Anzahl von Aufträgen im Spooler:** gibt die Anzahl von Aufträgen an, die sich im Spooler für den ausgewählten Drucker befinden.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.11.7. Etikett im Drucker speichern

> **PRODUKTEBENEN-INFO**
> Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Diese Aktion speichert eine Etikettenvorlage im Druckerspeicher. Diese Aktion ist ein wichtiger Teil des Speichern/Abrufen-Druckmodus, bei dem eine Etikettenvorlage zuerst im Druckerspeicher abgelegt und später daraus abgerufen wird. Die nicht veränderbaren Teile des Etikettendesigns sind bereits im Drucker gespeichert, sodass Sie nur die Daten für variable Etikettenobjekte zum Druckzeitpunkt angeben müssen. Weitere Informationen finden Sie im Abschnitt „Speichern/Abrufen-Druckmodus verwenden“ im NiceLabel Automation Benutzerhandbuch.

ANMERKUNG

Die erforderliche Zeit für die Übertragung der Etikettendaten wird erheblich verringert, da weniger Daten gesendet werden müssen. Diese Aktion wird häufig für Standalone-Druckumgebungen eingesetzt: Das Etikett wird im Drucker oder Applikator in der Produktionslinie gespeichert und später durch einen Software- oder Hardware-Trigger abgerufen, z. B. durch einen Barcode-Scanner oder eine Fotozelle.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Erweiterte Optionen für die Speicherung des Etiketts im Drucker** ermöglicht es Ihnen, ein Etikett und die bevorzugte Speichervariante auszuwählen.

- **Etikettenname zur Verwendung auf dem Drucker**: gibt den Namen an, der zum Speichern der Etikettenvorlage im Druckerspeicher verwendet werden soll. Geben Sie den Namen manuell ein oder aktivieren Sie **Datenquelle**, um den Namen dynamisch anhand einer vorhandenen oder neu erstellten Variablen zu definieren.

  WARNUNG

  Wenn das Etikett in einem Drucker gespeichert wird, sollte der Etikettenname in den erweiterten Optionen freigelassen werden. Dies verhindert Etikettennamen-Konflikte beim Abruf des Etiketts.

- **Variante speichern**: gibt den Speicherort im Drucker für gespeicherte Etikettenvorlagen an. Geben Sie den Speicherort manuell ein oder aktivieren Sie **Datenquelle**, um den Ort dynamisch anhand einer vorhandenen oder neu erstellten Variablen zu definieren.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (**`wahr`** oder **`falsch`**). Ist das Ergebnis des Ausdrucks **`wahr`**, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.11.8. PDF-Dokument drucken

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **LMS Enterprise**.

Die Aktion „PDF-Dokument drucken“ druckt statische PDF-Dokumente, die nicht mit den Etiketten in Ihren PowerForms Lösungen oder NiceLabel Automation Konfigurationen verbunden sind. Verwenden Sie diese Aktion, um PDF-Dokumente direkt aus Ihren Lösungen oder Konfigurationen heraus zu drucken. Die PDF-Dokumente können an folgenden Orten gespeichert werden:

- Ihr Computer
- NiceLabel Control Center
- Webserver
- Freigegebene Netzwerklaufwerke

TIPP

Die Aktion ist nützlich, wenn Sie planen, Ihre Verpackungen mit gedruckten PDF-Berichten zu enthaltenen Objekten auszustatten oder die Verpackungsdokumentation drucken möchten, ohne die Dateisuche zu öffnen.

ANMERKUNG

Wenn die Aktion „PDF-Dokument drucken“ verwendet wird, belegt sie einen Druckerplatz aus Ihrem Lizenzkontingent. Im NiceLabel Lizenzdokument finden Sie weitere Informationen zur Lizenzierung.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Drucker** gibt den Namen des für den aktuellen Druckauftrag zu verwendenden Druckers an.

- **Druckername**: Wählen Sie ihn aus der Liste lokal installierter Druckertreiber aus oder geben Sie einen Druckernamen ein. Wählen Sie **Datenquelle** aus, um den Drucker anhand einer Variablen dynamisch auszuwählen. Ist diese Option aktiviert, wählen Sie eine Variable aus (oder erstellen Sie eine Variable), die den bei Ausführung der Aktion genutzten Druckernamen enthält.

Die Gruppe **Datei** legt die Datei fest, an die umgeleitet wird.

- **Dateiname**: gibt an, welches PDF gedruckt werden soll.

ANMERKUNG

Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.12. Variablen

5.12.1. Variable einstellen

Diese Aktion weist der ausgewählten Variablen einen neuen Wert zu.

Variablen beziehen ihre Werte normalerweise von der Aktion „Datenfilter verwenden“ (verfügbar in Automation Builder), die Felder aus den empfangenen Daten extrahiert und sie Variablen zuordnet). Unter Umständen müssen Sie die Variablenwerte aber auch selbst einstellen, meistens zu Zwecken der Fehlerbehebung. In Automation Builder werden die Variablenwerte zwischen einzelnen Triggern nicht gespeichert, aber beibehalten, solange ein und derselbe Trigger verarbeitet wird.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die **Variable**-Gruppe definiert den Variablennamen und den Variablenwert.

- **Name**: Name der Variablen, die den geänderten Wert speichern soll.
- **Wert:** Wert, der für eine Variable eingestellt werden soll. Er kann entweder manuell oder durch eine vorhandene oder neu erstellte Variable dynamisch definiert werden.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.12.2. Variable Daten speichern

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Diese Aktion speichert Werte einer oder mehrerer Variablen in einer verbundenen Datendatei.

Im Modul NiceLabel Automation ermöglicht diese Aktion einen Datenaustausch zwischen Triggern. Um die Daten zurück in den Trigger einzulesen, verwenden Sie die Aktion „Variable Daten laden“.

TIPP

Die Werte werden in einer CSV-Datei gespeichert, deren erste Zeile Variablennamen enthält. Wenn die Variablen mehrzeilige Werte enthalten, werden die Zeichen für Zeilenwechsel (CR/LF) als `\n\r` codiert.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Einstellungen** definiert den Dateinamen.

- **Dateiname**: Datendatei, in der die variablen Daten gespeichert werden. Wenn der Name fest codiert ist, werden die Werte jedes Mal in derselben Datendatei gespeichert.

Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch.

Die Gruppe **Falls Datei existiert** bietet weitere Optionen zum Speichern der Werte.

- **Überschreibe Datei**: überschreibt die vorhandenen Daten mit neuen variablen Daten. Dabei gehen die alten Inhalte verloren.
- **Füge Daten an Datei**: fügt den vorhandenen Datendateien die Variablenwerte hinzu.

Die Gruppe **Dateistruktur** legt die Parameter für variable CSV-Datendateien fest:

- **Trennzeichen**: gibt die Art von Trennzeichen an (Tab, Semikolon, Komma oder benutzerdefiniertes Zeichen). Das Trennzeichen ist ein Zeichen, das die gespeicherten Variablenwerte voneinander trennt.
- **Textbegrenzer**: gibt das Zeichen an, das den gespeicherten Inhalt als Text markiert.
- **Datei-Codierung**: gibt die in der Datendatei zu verwendende Zeichencodierung an. **Auto** legt die Codierung automatisch fest. Falls nötig, kann der bevorzugte Codierungstyp aus der Dropdown-Liste ausgewählt werden.

  TIPP
  Als Standardauswahl bietet sich UTF-8 an.

- **Namen von Variablen in der ersten Reihe hinzufügen**: platziert den Variablennamen in der ersten Zeile der Datei.

Die Gruppe **Variablen** definiert die Variablen, deren Wert aus der Datendatei gelesen werden soll. Werte der vorhandenen Variablen werden mit den Werten aus der Datei überschrieben.

- **Alle Variablen**: variable Daten aus allen Variablen der Datendatei werden ausgelesen.
- **Ausgewählte Variablen**: variable Daten der aufgeführten Variablen werden aus der Datendatei gelesen.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.12.3. Variable Daten laden

> **PRODUKTEBENEN-INFO**
> Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Diese Aktion lädt Werte einer oder mehrerer Variablen aus der verbundenen Datendatei, welche mithilfe der Aktion **Variable Daten speichern** gespeichert wurde. Nutzen Sie diese Aktion, um die Daten zwischen Triggern auszutauschen. Sie können eine bestimmte Variable oder alle in der Datendatei gespeicherten Variablen laden.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.

- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Einstellungen** definiert den Dateinamen.

- **Dateiname**: gibt die Datei an, aus der die variablen Daten geladen werden sollen. Wenn der Name fest codiert ist, werden die Werte jedes Mal aus derselben Datei geladen.
  Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch.

Die Gruppeneinstellungen **Dateistruktur** müssen die Struktur der gespeicherten Datei aus der Aktion Variable Daten speichern widerspiegeln.

- **Trennzeichen**: gibt die Art von Trennzeichen an (Tab, Semikolon, Komma oder benutzerdefiniertes Zeichen). Das Trennzeichen ist ein Zeichen, das die Werte voneinander trennt.
- **Textbegrenzer**: gibt das Zeichen an, das Inhalt als Text markiert.
- **Datei-Codierung**: gibt die in der Datendatei verwendete Zeichencodierung an. **Auto** legt die Codierung automatisch fest. Wählen Sie, falls nötig, den bevorzugten Codierungstyp aus der Dropdown-Liste aus.

  TIPP
  Als Standardauswahl bietet sich UTF-8 an.

Die Gruppe **Variablen** definiert die Variablen, deren Werte aus der Datendatei geladen werden sollen.

- **Alle Variablen**: gibt an, dass alle in der Datendatei definierten Variablen gelesen werden sollen.
- **Ausgewählte Variablen**: gibt die Auswahl von einzelnen Variablen an, die aus der Datendatei gelesen werden sollen.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.12.4. Zeichenfolgenmanipulation

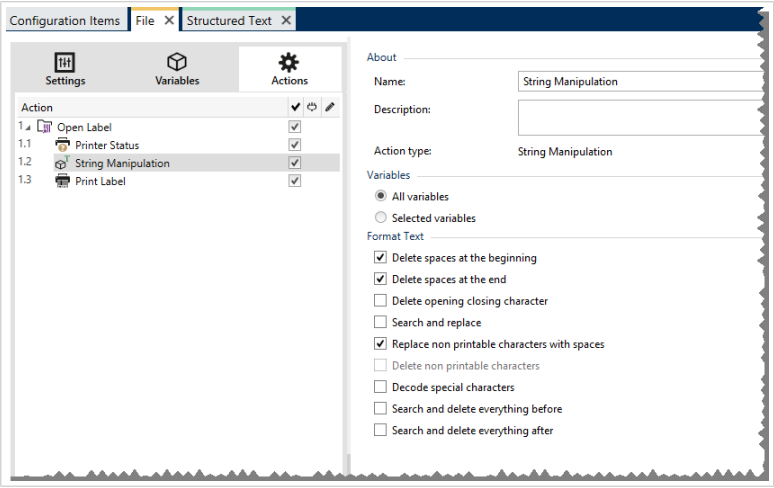
Diese Aktion legt fest, wie die Werte ausgewählter Variablen definiert werden sollen.

Die häufigsten Aktionen zur Bearbeitung von Zeichenfolgen sind: Führende und nachfolgende Leerzeichen löschen, Zeichen suchen und ersetzen, einführende und abschließende Anführungszeichen löschen.

Diese Funktion wird oft benötigt, wenn ein Trigger eine unstrukturierte Datendatei oder Altdaten empfängt. In solchen Fällen müssen die Daten anhand des Filters für **Unstrukturierte Daten** geparst werden. Mithilfe der Aktion „Zeichenfolgenmanipulation“ können Sie Anpassungen am Datenwert vornehmen.

ANMERKUNG

Wenn diese Aktion nicht genügend Bearbeitungsleistung für einen bestimmten Fall bietet, verwenden Sie stattdessen die Aktion **Skript ausführen**, um Ihre Daten anhand von Visual Basic- oder Python-Skripten zu bearbeiten.



Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Variablen** definiert die Variablen, deren Werte formatiert werden müssen.

- **Alle Variablen**: gibt an, dass alle in einer Datendatei definierten Variablen formatiert werden sollen.
- **Ausgewählte Variablen**: gibt eine Auswahl von Variablen aus der Datendatei an, die formatiert werden sollen.

Die Gruppe **Text formatieren** definiert die Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Es können mehrere Funktionen verwendet werden. Die Funktionen werden in der Reihenfolge ausgeführt, die im Editor angezeigt wird – von oben nach unten.

- **Leerzeichen am Anfang löschen**: löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Anfang einer Zeichenfolge.
- **Leerzeichen am Ende löschen**: löscht alle Leerzeichen (ASCII-Dezimalwert 32) am Ende einer Zeichenfolge.
- **Eröffnungs- und Abschlusszeichen löschen**: löscht die erste Instanz der ausgewählten Eröffnungs- und Abschlusszeichen, die in der Zeichenfolge enthalten sind.

Beispiel

Wenn „{“ als Eröffnungszeichen und „}“ als Abschlusszeichen verwendet wird, wird`{{selection}}` in `{selection}` konvertiert.

- **Suchen und ersetzen**: führt anhand der angegebenen Werte für Finde und Ersetzen durch eine Standardfunktion für Suchen und Ersetzen durch. Es werden reguläre Ausdrücke unterstützt.

> **ANMERKUNG**
> Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. NiceLabel 10Nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im Knowledge Base-Artikel.

- **Nicht-druckbare Zeichen durch Leerzeichen ersetzen**: ersetzt alle Steuerzeichen in einer Zeichenfolge durch Leerzeichen (ASCII-Dezimalwert 32). Nicht druckbare Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Nicht druckbare Zeichen löschen**: löscht alle Steuerzeichen in der Zeichenfolge. Nicht druckbare Zeichen sind Zeichen mit ASCII-Dezimalwerten zwischen 0 und 31 sowie 127 und 159.
- **Sonderzeichen dekodieren:** dekodiert Sonderzeichen (oder Steuercodes), die keine Entsprechung auf der Tastatur haben, z. B. Wagenrücklauf oder Zeilenvorschub.NiceLabel 10 verwendet eine menschenlesbare Codierung für solche Zeichen, beispielsweise <CR> für Wagenrücklauf und <LF> für Zeilenvorschub. Diese Option konvertiert Sonderzeichen aus der NiceLabel-Syntax in tatsächliche Binärzeichen.

> **Beispiel**
> Wenn Sie die Daten „<CR><LF>“ empfangen, fasst Designer sie als reine Zeichenfolge aus 8 Zeichen auf. Um die empfangenen Daten als zwei Binärzeichen zu erkennen, CR (Wagenrücklauf – ASCII-Code 13) und LF (Zeilenvorschub – ASCII-Code 10), müssen Sie diese Option aktivieren.

- **Suchen und Löschen von allem vor**: findet die angegebene Zeichenfolge und löscht alle Zeichen vor der definierten Zeichenfolge. Auch die Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach**: findet die angegebene Zeichenfolge und löscht alle Zeichen nach der definierten Zeichenfolge. Auch die Zeichenfolge kann gelöscht werden.
- **Großschreibung ändern**: Ändert alle Zeichen in Ihren Zeichenfolgen zu Groß- oder Kleinschreibung.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.

- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.13. Stapeldruck

5.13.1. FOR Schleife

PRODUKTEBENEN-INFO

Die hier beschriebene Funktion steht in **NiceLabel LMS Enterprise** zur Verfügung.

Die Aktion führt alle untergeordneten (geschachtelten) Aktionen mehrmals aus. Alle geschachtelten Aktionen werden in einer Schleife so häufig ausgeführt, wie durch die Differenz zwischen dem Start- und Endwert vorgegeben.

ANMERKUNG

Die Aktion „FOR Schleife“ leitet den Sitzungsdruckmodus ein – einen Druckoptimierungsmodus, der alle Etiketten anhand einer einzelnen Druckauftragsdatei in einer Schleife druckt. Weitere Informationen finden Sie im Abschnitt „Sitzungsdruck“ des NiceLabel Automation Benutzerhandbuchs.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Schleifeneinstellungen** enthält die folgenden Optionen:

- **Startwert**: Referenz für den Startpunkt der Schleife. Wählen Sie Datenquelle, um den Startwert dynamisch anhand eines Variablenwerts zu definieren. Wählen Sie für den Start eine Variable mit einem numerischen Wert aus oder erstellen Sie eine solche Variable.
- **Endwert**: Referenz für den Endpunkt. Wählen Sie Datenquelle, um den Startwert dynamisch anhand eines Variablenwerts zu definieren. Wählen Sie für den Start eine Variable mit einem numerischen Wert aus oder erstellen Sie eine solche Variable.

  ### TIPP

  Für **Startwert** und **Endwert** sind negative Werte erlaubt.

- **Loop-Wert in Variable speichern**: speichert den Wert des aktuellen Schrittes der Schleife in einer vorhandenen oder neu erstellten Variablen. Der Schleifen-Schrittwert kann einen beliebigen Wert zwischen dem Start- und Endwert enthalten. Speichern Sie den Wert, um ihn in einer anderen Aktion zur Erkennung der aktuellen Iteration zu verwenden.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.13.2. Datenfilter verwenden

Diese Aktion wendet die Filterregeln auf die Eingabedatenquelle an. Als Ergebnis der Aktion werden Felder aus Eingabedaten extrahiert und ihre Werte den verbundenen Variablen zugeordnet.

Die Aktion „Datenfilter verwenden“ führt also den ausgewählten Filter aus und weist die Werte den entsprechenden Variablen zu.

- **Elemente auf niedrigerer Ebene:** Die Aktion kann Elemente auf niedrigerer Ebene erstellen, die durch **„Für jede Zeile"** oder **„Für jeden Datenblock in ..."** identifiziert werden. In diesem Fall extrahiert der Filter die Daten nicht auf Dokumentebene (mit fest codierter Feldpositionierung), sondern relativ aus den Unterbereichen, die sich wiederholende Abschnitte enthalten. Stellen Sie dabei sicher, dass Sie Ihre Aktionen unter solchen Elementen anordnen. Die Aktionen müssen unter solchen Elementen geschachtelt werden.
- **Variablen Feldern zuordnen:** Die Zuordnung von Trigger-Variablen und Filterfeldern wird entweder manuell definiert oder erfolgt automatisiert, je nachdem, wie der Filter konfiguriert ist. Falls Sie manuell definierte Felder im Filter haben, müssen Sie die Felder auch manuell der entsprechenden Variablen zuordnen.

  ANMERKUNG

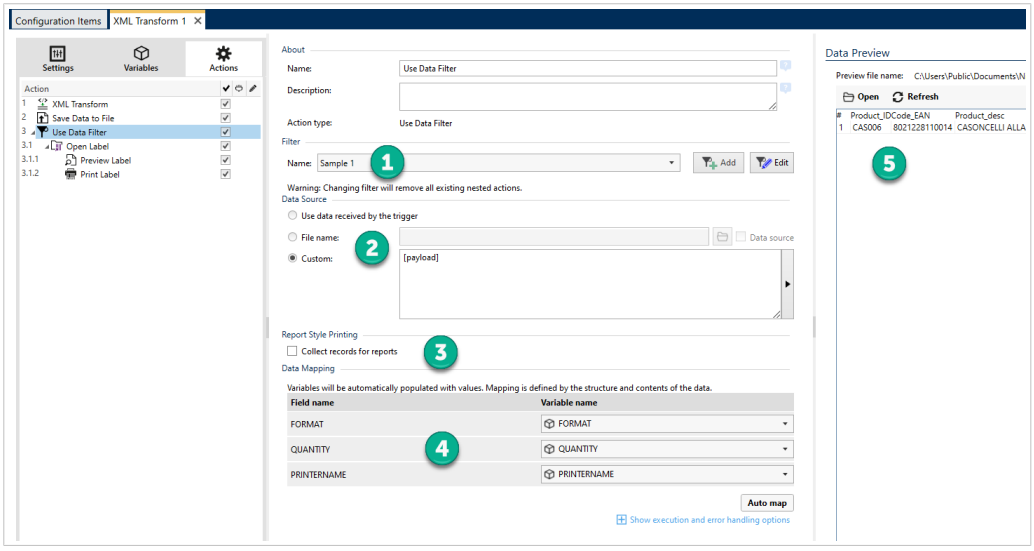
  Es empfiehlt sich, Felder mit Namen zu definieren, die mit den Namen der Etikettenvariablen identisch sind. In diesem Fall erfolgt nach Klicken auf die Schaltfläche **Automatisch verbinden** eine automatische Zuordnung identischer Namen.

- **Ausführung des Filters testen:** Wenn die Zuordnung von Variablen zu Feldern abgeschlossen ist, können Sie die Ausführung des Filters testen. Das Ergebnis wird auf dem Bildschirm in einer Tabelle angezeigt. Die Anzahl von Zeilen in der Tabelle gibt die Häufigkeit der Ausführung von Aktionen auf der ausgewählten Ebene an. Die Spaltennamen entsprechen den Variablennamen. Die Zellen enthalten die Werte, die der jeweiligen Variablen durch den Filter zugewiesen wurden. Der Standardname der Vorschaudatei wird aus der Filterdefinition abgeleitet; Sie können den Filter auch auf jede andere Datei anwenden.
- **Datensätze für Berichte erfassen** dient zur Erfassung Ihrer Daten, damit Sie Datenfilter zum Erstellen von Berichten verwenden können. Weitere Informationen finden Sie unter Abschnitt 9.7, „Berichte automatisieren".

  PRODUKTEBENEN-INFO

  Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .



Auf der Benutzeroberfläche Ihres Datenfilters navigieren.

1. Filterauswahl
2. Daten, die Ihr Filter zum Ausführen der Regeln verwendet.
3. Erfassung von Berichtdatensätzen.
4. Zuordnung Ihrer Felder (aus dem Filter) auf Variablen (aus dem Etikett/Trigger).
5. Filterausführungs-Vorschau.

Weitere Informationen finden Sie in den Abschnitten „Informationen zu Filtern“ und „Beispiele“ im NiceLabel Automation Benutzerhandbuch.

Die Gruppe **Filter** ermöglicht es Ihnen, auszuwählen, welcher Filter verwendet werden soll.

- **Name:** gibt den Namen des anzuwendenden Filters an. Er kann entweder fest codiert oder durch eine vorhandene oder neu erstellte Variable dynamisch definiert werden. Diese Liste enthält alle in der aktuellen Konfiguration definierten Filter. Sie können die unteren drei Elemente in der Liste verwenden, um einen neuen Filter zu erstellen.

ANMERKUNG

Bei Auswahl eines anderen Filters werden alle unter dieser Aktion geschachtelten Aktionen entfernt. Wenn Sie die aktuell definierten Aktionen beibehalten möchten, bewegen Sie sie aus der Aktion **Datenfilter verwenden**. Wenn Aktionen aus Versehen verloren gehen, können Sie Ihre Aktion **Rückgängig** machen und zur vorherigen Konfiguration zurückkehren.

Die Gruppe **Datenquelle** ermöglicht es Ihnen, die Inhalte festzulegen, die an den Drucker gesendet werden sollen.

- **Vom Trigger empfangene Daten verwenden:** wählt die vom Trigger empfangenen Daten zur Verwendung in einem Filter aus. In diesem Fall nutzt die Aktion die ursprünglichen, vom Trigger empfangenen Daten und wendet die Filterregeln auf sie an.

Beispiel

Wenn Sie einen Datei-Trigger verwenden, stellen die Daten Inhalte der überwachten Datei dar. Wenn Sie einen Datenbank-Trigger verwenden, sind die Daten ein von der Datenbank ausgegebenes Daten-Set. Wenn Sie einen TCP/IP-Trigger verwenden, sind es Rohdaten, die über ein Socket empfangen wurden.

- **Dateiname:** legt den Speicherort und den Dateinamen der Datei fest, welche die Daten enthält, auf die die Filterregeln angewandt werden. Der Inhalt der angegebenen Datei wird vom Filter verwendet. Die Option **Datenquelle** aktiviert den variablen Dateinamen. Sie müssen eine Variable auswählen oder erstellen, die den Pfad und/oder Dateinamen enthält.
- **Benutzerdefiniert:** gibt den benutzerdefinierten Inhalt an, der vom Filter geparst werden soll. Sie können festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzufügen, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie eine Variable aus der Liste ein. Weitere Informationen finden Sie im Abschnitt „Zusammengesetzte Werte verwenden“ im NiceLabel Automation Benutzerhandbuch.

Das Feld **Datenvorschau** bietet eine Übersicht über den Prozess der Filterausführung, nachdem der Inhalt der für die Voransicht ausgewählten Datei ausgelesen und der ausgewählte Filter auf sie angewandt wurde.

Die Regeln im Filter extrahieren Felder. Die Tabelle zeigt das Ergebnis der Extraktion an. Jede Zeile in der Tabelle beinhaltet Daten für ein einzelnes Etikett. Jede Spalte steht für eine Variable.

Um ein Ergebnis sehen zu können, konfigurieren Sie die Zuordnung von Feldern zu passenden Variablen. Abhängig von der Filterdefinition kann die Zuordnung von Feldern und Variablen manuell oder automatisch erfolgen.

- **Name der Vorschaudatei:** Gibt die Datei an, welche die Daten enthält, die durch den Filter geparst werden sollen. Die Vorschaudatei wird aus der Filterdefinition kopiert. Wenn Sie den Namen der Vorschaudatei ändern, wird der neue Name gespeichert.
- **Öffnen:** wählt eine andere Datei aus, auf die die Filterregeln angewandt werden sollen.
- **Aktualisieren:** wendet die Filterregeln erneut auf den Inhalt der Vorschaudatei an. Das Feld **Datenvorschau** wird mit dem neuen Ergebnis aktualisiert.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.

- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.13.3. Für jeden Datensatz

Diese Aktion führt untergeordnete geschachtelte Aktionen mehrmals aus. Alle geschachtelten Aktionen werden als Schleife für jeden Datensatz in der Eingabemasken-Tabelle mit verbundener Datenbank ausgeführt.

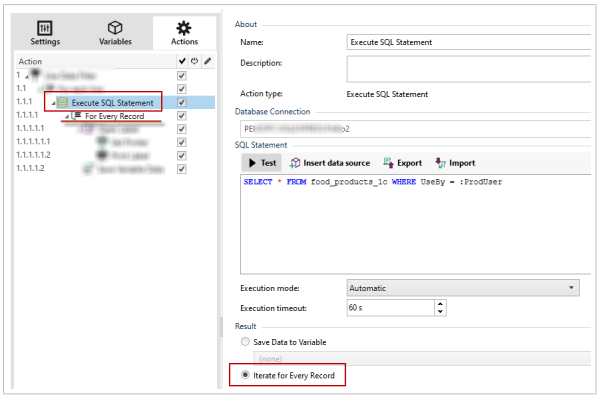
Die Über-Gruppe gibt die ausgewählte Aktion an.

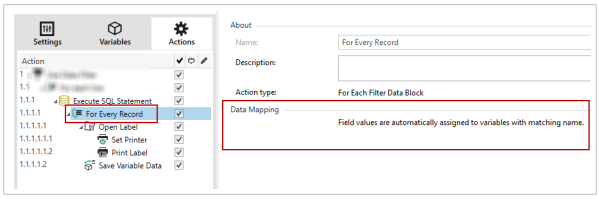
- Name: ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- Beschreibung: benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- Aktionstyp: Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Einstellungen** wählt die Datensätze aus.

- **Masken-Tabelle**: Masken-Tabelle, die Datensätze enthält, für die eine Aktion wiederholt werden soll.
- **Alle Datensätze verwenden**: wiederholt eine Aktion für alle Datensätze in einer definierten Tabelle.
- **Ausgewählte Datensätze verwenden**: wiederholt eine Aktion nur für die ausgewählten Datensätze.

Wenn Sie die Aktion verwenden **SQL-Anweisung ausführen** mit aktivierter Option **Iteriere für jeden Datensatz**, NiceLabel fügt sich automatisch ein **Für jeden Rekord** Handlung. Es erscheint ein Hinweis zur automatischen Zuordnung Ihrer Variablen.





Eingabeaufforderungsvariablen auf Ihrem Etikett verbinden sich automatisch mit Ihren Datenbankfeldern mit denselben Namen. Lesen Sie die Anweisungen zum Erstellen Ihrer Lösungen:

WICHTIG

Wenn Sie Ihre Lösung mit einer Datenbankverbindung erstellen, verwenden Sie anstelle von Datenbankfeldern Eingabeaufforderungsvariablen auf Ihren Etiketten.

Verwenden Sie für Eingabeaufforderungsvariablen dieselben Namen wie für Datenbankfelder definiert, zum Beispiel:

Datenbankfeld: `food_products_1c.ProdCode`

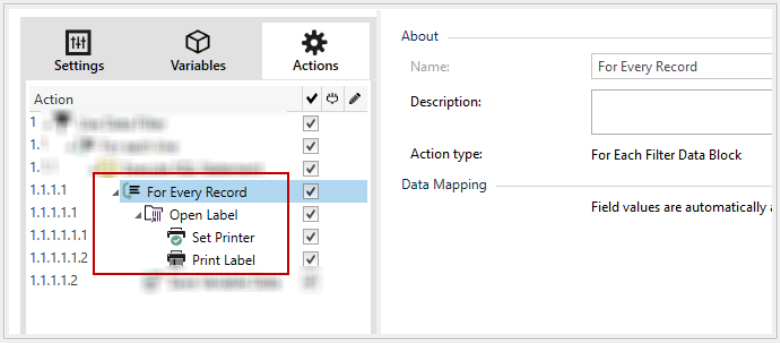
Prompt-Variable auf Ihrem Etikett: `ProdCode`

NiceLabel ordnet dann automatisch entsprechende Variablen mit Datenbankfeldern zu.

Beispiel

Ihre Lösung ist mit Ihrer Datenbank verbunden. Ihre Konfiguration filtert angeforderte Datenbankeinträge und Sie möchten gefilterte Einträge auf Ihre Etiketten drucken.

Aktion verwenden **Für jeden Rekord** und verschachteln Sie Aktionen, um Ihre Etiketten zu drucken.



Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

> **ANMERKUNG**
> Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.14. Daten und Konnektivität

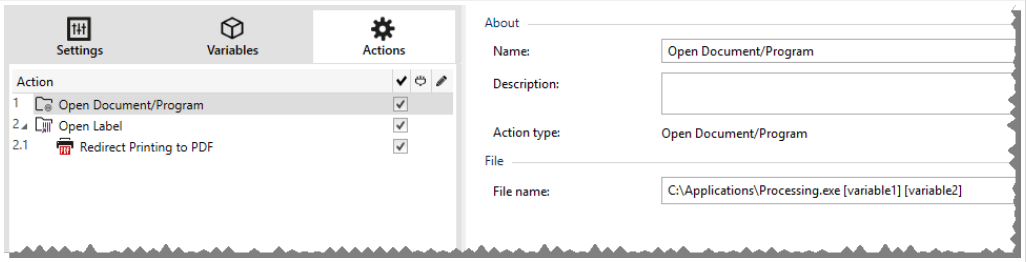
5.14.1. Dokument/Programm öffnen

Diese Aktion stellt eine Schnittstelle mit einer externen Anwendung bereit und öffnet sie anhand einer Befehlszeileneingabe.

Externe Anwendungen können zusätzliche Verarbeitungsschritte ausführen und die Ergebnisse wieder an NiceLabel 10 ausgeben. Mithilfe dieser Aktion kann es sich mit Software von Drittanbietern verbinden, in der zusätzliche Datenverarbeitungsschritte ausgeführt oder Daten abgerufen werden können. Die externe Software kann Datenantworten bereitstellen, indem sie sie als Datei speichert, aus der Sie die Daten in Variablen einlesen können.

Sie können die Werte der Variablen an das Programm zurückgeben, indem Sie sie in der Befehlszeile in eckigen Klammern auflisten.

```
C:\Applications\Processing.exe [variable1] [variable2]
```



ANMERKUNG

Wenn Sie diese Aktion in NiceLabel 10 Lösungen verwenden, können Sie direkt aus Ihren Eingabemasken heraus Webseiten öffnen oder E-Mail-Nachrichten erstellen. Siehe Abschnitt „Auf der Maske Hyperlinks erstellen und E-Mails versenden“ im NiceLabel 10 Benutzerhandbuch.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.

- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Datei** definiert die zu öffnende Datei.

- **Dateiname**: Speicherort und Dateiname der zu öffnenden Datei oder Anwendung.
  Der ausgewählte Dateiname kann fest codiert werden, wodurch jedes Mal dieselbe Datei verwendet wird. Wenn nur ein Dateiname ohne Pfad angegeben ist, wird der Ordner mit NiceLabel Automation Konfigurationsdatei (.MISX) verwendet. Sie können eine relative Referenz auf den Dateinamen verwenden, in der der Ordner mit .MISX-Datei als Hauptverzeichnis verwendet wird.
  Datenquelle: aktiviert den variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und/oder Dateinamen enthält, oder kombinieren Sie mehrere Variablen, die den Dateinamen bilden. Weitere Informationen finden Sie im Abschnitt Zusammengesetzte Werte verwenden im NiceLabel Automation Benutzerhandbuch.

  ANMERKUNG
  Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt „Zugriff auf freigegebene Ressourcen im Netzwerk“ im NiceLabel Automation Benutzerhandbuch.

Die Gruppe **Ausführungsoptionen** definiert Optionen zum Öffnen von Programmen.

- **Fenster ausblenden**: macht das Fenster des geöffneten Programms unsichtbar. Da NiceLabel 10 als Dienstanwendung innerhalb einer eigenen Sitzung ausgeführt wird, kann es nicht mit dem Desktop interagieren, selbst wenn es mit den Berechtigungen des aktuell angemeldeten Benutzers ausgeführt wird. Aus Sicherheitsgründen hat Microsoft solche Interaktionen in Windows Vista und neueren Betriebssystemen unterbunden.
- **Auf Fertigstellung warten**: gibt vor, dass die Aktionsausführung auf den Abschluss dieser Aktion wartet, bevor sie mit der nächsten geplanten Aktion fortfährt.

  TIPP
  Aktivieren Sie diese Option, wenn die folgende Aktion von den Ergebnissen aus der externen Anwendung abhängt.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.

- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.14.2. Daten in Datei speichern

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Diese Aktion speichert Variablenwerte oder andere Datenströme (wie z. B. Binärdaten) in einer ausgewählten Datei. Der NiceLabel Automation-Dienst muss Schreibzugriff auf den angegebenen Ordner haben.

Die Gruppe **Datei** definiert die zu öffnende Datei.

- **Dateiname**: Speicherort der Datei, die innerhalb dieser Aktion geöffnet werden soll.
  Pfad und Dateiname können fest codiert werden, und jedes Mal wird dieselbe Datei verwendet. Wenn nur ein Dateiname ohne Pfad angegeben ist, wird der Ordner mit NiceLabel Automation

Konfigurationsdatei (.MISX) verwendet. Sie können eine relative Referenz auf den Dateinamen verwenden, in der der Ordner mit .MISX-Datei als Hauptverzeichnis verwendet wird.

Datenquelle: aktiviert den variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und/oder Dateinamen enthält, oder kombinieren Sie mehrere Variablen, die den Dateinamen bilden. Weitere Informationen finden Sie im Abschnitt „Zusammengesetzte Werte verwenden“ im NiceLabel Automation Benutzerhandbuch.

Die Gruppe **Falls Datei existiert** legt die Optionen für den Fall fest, dass die Datei bereits vorhanden ist.

- **Überschreibe Datei**: überschreibt vorhandene Daten mit neuen Daten. Dabei gehen die alten Inhalte verloren.
- **Füge Daten an Datei**: fügt den vorhandenen Datendateien Variablenwerte hinzu.

Die Gruppe **Inhalt** legt fest, welche Dateien in die festgelegte Datei geschrieben werden sollen.

- **Vom Trigger empfangene Daten verwenden:** vom Trigger empfangene Originaldaten werden in der Datei gespeichert. Diese Option führt im Endeffekt zu einer exakten Kopie der eingehenden Daten.
- **Benutzerdefiniert**: speichert den Inhalt so, wie er im Textbereich angegeben ist. Es sind Festwerte, variable Werte und Sonderzeichen erlaubt. Um Variablen und Sonderzeichen einzugeben, klicken Sie auf die Pfeil-Schaltfläche rechts neben dem Textbereich. Weitere Informationen finden Sie im Abschnitt „Werte in einem Objekt kombinieren“ des NiceLabel Automation Benutzerhandbuchs.
- **Codieren**: Codierungstyp für die gesendeten Daten. **Auto** legt die Codierung automatisch fest. Wählen Sie, falls nötig, den bevorzugten Codierungstyp aus der Dropdown-Liste aus.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.14.3. Daten aus Datei lesen

PRODUKTEBENEN-INFO

Die beschriebene Funktion ist in **NiceLabel LMS Enterprise** und **NiceLabel LMS Pro** verfügbar.

Die Aktion liest den Inhalt der angegebenen Datei und speichert ihn in einer Variablen. Der Inhalt aller Dateitypen, einschließlich Binärdaten, kann gelesen werden.

Normalerweise empfängt das Modul Automation Builder Daten für den Etikettendruck anhand eines Triggers. Zum Beispiel: Wenn ein Dateitrigger verwendet wird, wird der Inhalt der Triggerdatei automatisch gelesen und kann durch Filter geparst werden. Möglicherweise wollen Sie die Filter jedoch umgehen, um externe Daten zu erhalten. Nachdem Sie diese Aktion ausgeführt und die Daten in einer Variablen gespeichert haben, können Sie zur Nutzung der Daten jede verfügbare Aktion verwenden.

Diese Aktion ist nützlich:

- Wenn Sie vom Trigger empfangene Daten mit Daten verbinden wollen, die in einer Datei gespeichert sind.

> **WARNUNG**
> Wenn Sie Daten aus Binärdateien laden (z. B. Bitmap-Bilder oder Druckdateien), müssen Sie sicherstellen, dass die Variable zum Speichern des ausgelesenen Inhalts als **binäre Variable** definiert ist.

- Wenn Sie Daten zwischen Triggern austauschen möchten. Ein Trigger bereitet die Daten vor und speichert sie als Datei (anhand der Aktion Daten in Datei speichern.

Die Gruppe **Datei** definiert die Datei, aus der die Daten gelesen werden sollen.

- **Dateiname**: Speicherort der Datei, die innerhalb dieser Aktion ausgelesen werden soll.
  Pfad und Dateiname können fest codiert werden, und jedes Mal wird dieselbe Datei verwendet. Wenn nur ein Dateiname ohne Pfad angegeben ist, wird der Ordner mit NiceLabel Automation Konfigurationsdatei (.MISX) verwendet. Sie können eine relative Referenz auf den Dateinamen verwenden, in der der Ordner mit .MISX-Datei als Hauptverzeichnis verwendet wird.
  Datenquelle: aktiviert den variablen Dateinamen. Wählen Sie eine Variable aus, die den Pfad und/oder Dateinamen enthält, oder kombinieren Sie mehrere Variablen, die den Dateinamen bilden. Weitere Informationen finden Sie im Abschnitt Zusammengesetzte Werte verwenden im NiceLabel Automation Benutzerhandbuch.

  > **ANMERKUNG**
  > Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt „Zugriff auf freigegebene Ressourcen im Netzwerk“ im NiceLabel Automation Benutzerhandbuch.

Die **Inhalt**-Gruppe legt die Details in Bezug auf den Dateiinhalt fest.

- **Variable**: Variable, die den Dateiinhalt speichert. Es sollte mindestens eine Variable (vorhanden oder neu erstellt) definiert werden.
- **Codieren**: Codierungstyp für die gesendeten Daten. **Auto** legt die Codierung automatisch fest. Wählen Sie, falls nötig, den bevorzugten Codierungstyp aus der Dropdown-Liste aus.

  > **ANMERKUNG**
  > „Codieren“ kann nicht ausgewählt werden, wenn die Daten aus einer binären Variablen gelesen werden. In diesem Fall enthält die Variable die Daten, wie sie vorliegen.

Die Gruppe **Bei Fehler wiederholen** definiert, wie die Ausführung der Aktion fortfahren soll, falls der Zugriff auf die angegebene Datei fehlschlägt.

TIPP

Das Modul Automation Builder ist vielleicht nicht in der Lage, auf die Datei zuzugreifen, da sie durch eine andere Anwendung gesperrt wird. Wenn eine andere Anwendung Daten in die ausgewählte Datei schreibt oder die Datei im exklusiven Modus geöffnet hat, kann sie nicht gleichzeitig von einer anderen Anwendung geöffnet werden, nicht einmal zum Lesen. Andere mögliche Ursachen für erneute Versuche zum Ausführen der Aktion sind: Datei existiert (noch) nicht, Ordner existiert (noch) nicht oder der Dienstbenutzer hat keine Berechtigung zum Zugriff auf die Datei.

- **Wiederholungsversuche**: definiert die Anzahl von wiederholten Versuchen, auf die Datei zuzugreifen. Ist der Wert auf 0 eingestellt, wird kein wiederholter Versuch unternommen.
- **Wiederholungsintervall**: Zeitintervall zwischen einzelnen Wiederholungsversuchen in Millisekunden.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung

angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.14.4. Datei löschen

PRODUKTEBENEN-INFO

Die beschriebene Funktion ist in **NiceLabel LMS Enterprise** und **NiceLabel LMS Pro** verfügbar.

Diese Aktion löscht eine ausgewählte Datei von einer Festplatte.

Das Modul NiceLabel Automation wird unter einem bestimmten Windows-Benutzerkonto als Dienst ausgeführt. Stellen Sie sicher, dass dieses Konto über Berechtigungen zum Löschen der Datei in einem festgelegten Ordner verfügt.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die **Datei**-Gruppe legt die dateibezogenen Details fest.

- **Dateiname**: der Name der zu löschenden Datei. Der **Dateiname** kann fest codiert werden. **Datenquelle** definiert den **Dateinamen** anhand einer vorhandenen oder neu erstellten Variablen dynamisch.
  Pfad und Dateiname können fest codiert werden, und jedes Mal wird dieselbe Datei verwendet. Wenn nur ein Dateiname ohne Pfad angegeben ist, wird der Ordner mit NiceLabel Automation Konfigurationsdatei (.MISX) verwendet. Sie können eine relative Referenz auf den Dateinamen verwenden, in der der Ordner mit .MISX-Datei als Hauptverzeichnis verwendet wird.
  Die **Datenquelle**-Option aktiviert den variablen Dateinamen. Wählen Sie eine Variable aus (oder erstellen Sie eine), die den Pfad und/oder Dateinamen enthält, oder kombinieren Sie mehrere Variablen, die den Dateinamen bilden. Weitere Informationen finden Sie im Abschnitt Zusammengesetzte Werte verwenden im NiceLabel Automation Benutzerhandbuch.

> **ANMERKUNG**
> Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt „Zugriff auf freigegebene Ressourcen im Netzwerk“ im NiceLabel Automation Benutzerhandbuch.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.14.5. SQL-Anweisung ausführen

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Diese Aktion sendet SQL-Anweisungen an einen verbundenen SQL-Server und ruft die Ergebnisse ab. Die verfügbaren Befehle sind SELECT, INSERT, UPDATE und DELETE..

Verwenden Sie die Aktion „SQL-Anweisung ausführen“, um Folgendes zu erreichen:

- **Zusätzliche Daten aus einer Datenbank erhalten:** Im Modul Automation Builder empfängt ein Trigger Daten für den Etikettendruck, aber nicht alle erforderlichen Werte. Beispielsweise empfängt ein Trigger Werte für **`Product ID`** und **`Description`**, aber nicht für **`Price`**. Der Wert für **`Price`** muss in der SQL-Datenbank nachgeschlagen werden.
  Beispiel für SQL-Code:

  ```
  SELECT Price FROM Products
  WHERE ID = :[Product ID]
  ```

  Die **`ID`** ist ein Feld in der Datenbank, **`Product ID`** ist eine im Trigger definierte Variable.

- **Datensätze in einer Datenbank aktualisieren oder löschen:** Nachdem ein Etikett gedruckt wurde, aktualisieren Sie den Datensatz und senden Sie dem System ein Signal, dass der betreffende Datensatz bereits verarbeitet wurde.
  Beispiel für SQL-Code:
  Stellen Sie den Wert für das Tabellenfeld **`AlreadyPrinted`** für den soeben verarbeiteten Datensatz auf **`True`** ein.

  ```
  UPDATE Products
  SET AlreadyPrinted = True
  WHERE ID = :[Product ID]
  ```

  Oder löschen Sie den aktuellen Datensatz aus der Datenbank, da er nicht mehr benötigt wird.

  ```
  DELETE FROM Products
  WHERE ID = :[Product ID]
  ```

  Die **`ID`** ist ein Feld in der Datenbank, **`Product ID`** ist eine im Trigger definierte Variable.

ANMERKUNG

Um eine Variable in einer SQL-Anweisung zu nutzen, müssen Sie ihrem Namen einen Doppelpunkt (:) voranstellen. So geben Sie an, dass ein Variablenname folgt.

WICHTIG

Wenn Sie Ihre Lösung mit einer Datenbankverbindung erstellen, verwenden Sie anstelle von Datenbankfeldern Eingabeaufforderungsvariablen auf Ihren Etiketten.

Verwenden Sie für Eingabeaufforderungsvariablen dieselben Namen wie für Datenbankfelder definiert, zum Beispiel:

Datenbankfeld: `food_products_1c.ProdCode`

Prompt-Variable auf Ihrem Etikett: `ProdCode`

NiceLabel ordnet dann automatisch entsprechende Variablen mit Datenbankfeldern zu.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Datenbankverbindung** definiert die Datenbankverbindung, die für die Anweisung verwendet wird.

TIPP

Bevor Sie eine SQL-Anweisung an die Datenbank senden können, müssen Sie die Datenbankverbindung einrichten. Klicken Sie auf die Schaltfläche **Definieren** und folgen Sie den Anweisungen auf dem Bildschirm. Sie können nur eine Verbindung zu einer Datenquelle herstellen, die mithilfe von SQL-Befehlen gesteuert werden kann, und können daher keine Textdateien (CSV) oder Excel-Dateien verwenden.

Die Gruppe **SQL-Anweisung** definiert eine auszuführende SQL-Anweisung oder -Abfrage.

TIPP

Anweisungen in Data Manipulation Language (DML) sind erlaubt, um vorhandene Datenbanktabellen abzufragen.

Verwenden Sie Standard-SQL-Anweisungen wie SELECT, INSERT, DELETE und UPDATE, einschließlich Joins, Funktionen und Stichwörtern. Die Anweisungen in DDL-Sprache zur Erstellung von Datenbanken und Tabellen (CREATE DATABASE, CREATE TABLE) bzw. zum Löschen von ihnen (DROP TABLE) sind nicht erlaubt.

- **Test**: öffnet den Bereich **Datenvorschau**. Die simulierte Ausführung (standardmäßig ausgewählt) testet die Ausführung von SQL-Anweisungen. Klicken Sie auf **Ausführen**, um die Simulation auszuführen.

  ### TIPP

  Im Bereich **Datenvorschau** können Sie die Anwendung Ihrer SQL-Anweisung auf einen Satz von Echtzeitdaten testen. Um die Daten vor unbeabsichtigten Änderungen zu schützen, muss die Option **Ausführung simulieren** aktiviert sein. Die Anweisungen INSERT, DELETE und UPDATE werden ausgeführt. So können Sie eine Rückmeldung zur Anzahl der betroffenen Datensätze erhalten; danach werden alle Aktionen rückgängig gemacht.

  Wenn Sie Trigger-Variablen in der SQL-Anweisung verwenden, können Sie deren Werte für die Testausführung angeben.

- **Datenquelle einfügen**: fügt vordefinierte oder neu erstellte Variablen in eine SQL-Anweisung ein.
- **Exportieren/Importieren**: ermöglicht das Exportieren und Importieren von SQL-Anweisungen in eine/aus einer externe(n) Datei.
- **Ausführungsmodus**: gibt den exakten Modus der Ausführung der SQL-Anweisung an.

  ### TIPP

  Im Fall von komplexen SQL-Abfragen wird es zunehmend schwerer, automatisch zu bestimmen, worin die gewünschte Aktion bestehen soll. Falls die integrierte Logik Probleme damit hat, Ihre Absicht zu verstehen, wählen Sie die Hauptaktion manuell aus.

  - **Automatisch**: bestimmt die Aktion automatisch.
  - **Gibt einen Satz von Datensätzen aus (SELECT):** gibt das Daten-Set mit Datensätzen aus.
  - **Gibt keinen Satz von Datensätzen aus (INSERT, DELETE, UPDATE):**nutzen Sie diese Option, wenn Sie eine Abfrage ausführen, die keine Datensätze ausgibt. Sie fügen entweder neue Datensätze ein oder löschen bzw. aktualisieren die vorhandenen Datensätze. Das Ergebnis ist eine Statusantwort mit der Anzahl von Zeilen, auf die sich Ihre Abfrage ausgewirkt hat.

- **Zeitüberschreitung bei der Ausführung**: ermöglicht es Ihnen, die Zeitverzögerung für das Senden Ihrer Befehle an den SQL-Server festzulegen. Verwenden Sie die Zeitüberschreitung, wenn Sie mehrere aufeinander folgende SQL-Befehle senden, die eine längere Verarbeitungszeit erfordern. Geben Sie die gewünschte Zeitüberschreitungsdauer in Sekunden an. Standardmäßig beträgt die Zeitüberschreitungsdauer bei der Ausführung 60 Sekunden. Wenn Sie möchten, dass Ihr Datenbankanbieter die Zeitüberschreitung festlegt, geben Sie 0 Sekunden ein.

Die **Ergebnis**-Gruppe ermöglicht es Ihnen, festzulegen, wie die SQL-Anweisung gespeichert werden soll, und die Schritte für die Aktion zu definieren.

- **Daten in Variable speichern**: Wählt eine Variable aus (oder erstellt eine Variable), um das Ergebnis der SQL-Anweisung zu speichern. Diese Option hängt vom ausgewählten **Ausführungsmodus** ab.

- **Ergebnis der SELECT-Anweisung.** Nach Ausführung einer SELECT-Anweisung wird ein Daten-Set mit Datensätzen ausgegeben. Sie erhalten den Textinhalt im CSV-Format. Die erste Zeile enthält die Namen der im Ergebnis ausgegebenen Felder. Die nächste Zeile enthält Datensätze.

  > **ANMERKUNG**
  > Um die Werte aus dem ausgegebenen Daten-Set zu extrahieren und in anderen Aktionen zu verwenden, definieren Sie die Aktion „Datenfilter verwenden“ und wenden Sie sie auf den Inhalt dieser Variablen an (diese Aktion steht in Automation Builder zur Verfügung).

- **Ergebnis der INSERT-, DELETE- und UPDATE-Anweisungen.** Wenn Sie INSERT-, DELETE- und UPDATE-Anweisungen nutzen, erhalten Sie als Ergebnis die Anzahl betroffener Datensätze in der Tabelle.

- **Für jeden Datensatz iterieren**. Sofern aktiviert, wird automatisch eine neue Aktion „Für jeden Datensatz“ hinzugefügt. Alle geschachtelten Aktionen werden für jeden Datensatz wiederholt, der anhand der SQL-Anweisung ausgegeben wurde.

  > **ANMERKUNG**
  > Automatische Zuordnung ist aktiviert. Die Aktion „Für jeden Datensatz“ kann nicht gelöscht werden.
  >
  > Datenbank-Feld: `food_products_1c.ProdCode`
  >
  > Aufforderungsvariable auf Ihrem Etikett: `ProdCode`

Die Gruppe **Bei Fehler wiederholen** ermöglicht es Ihnen, die Aktion so zu konfigurieren, dass sie laufend versucht, eine Verbindung zu einem Datenbankserver herzustellen, wenn dies beim ersten Versuch nicht gelingt. Falls die Aktion nach der definierten Anzahl von Versuchen immer noch fehlschlägt, wird ein Fehler ausgegeben.

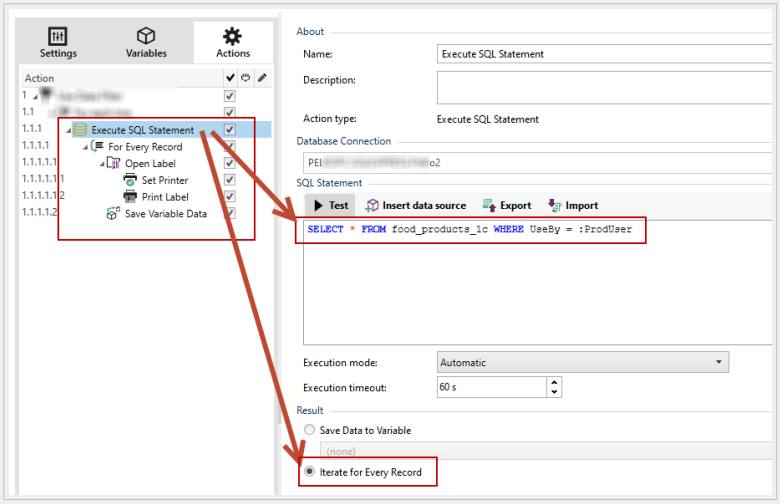
- **Wiederholungsversuche**: legt die Anzahl von Versuchen fest, eine Verbindung zum Datenbankserver herzustellen.
- **Wiederholungsintervall**: legt die Zeit zwischen einzelnen Wiederholungsversuchen fest.

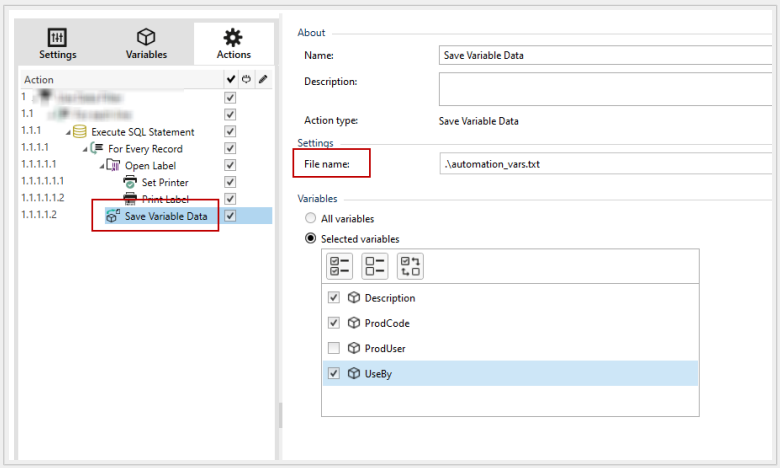
Beispiel

Sie möchten Etiketten mit Daten aus Ihrer Datenbank `food_products_1c` drucken, aber nur Datensätze mit dem vordefinierten Feldwert `UseBy` .

Sie definieren den `UseBy`-Wert mit der Variablen `ProdUser` . In diesem Fall ist der **ProdUser**-Wert "3".

Nachdem jedes Etikett gedruckt wurde, NiceLabel werden die Datenbankwerte in eine Textdatei auf Ihre Festplatte geschrieben. Verwenden Sie die folgenden Aktionen:





Es werden nur Etiketten gedruckt, deren `UseBy`-Wert gleich "3" ist, und die Werte werden in eine Textdatei geschrieben:

```
"Description";"ProdCode";"UseBy"
"Choux pastry";"977000000011";"3"
"Croline";"977000000012";"3"
"Danish pastry";"977000000013";"3"
"Kouign-amann";"977000000022";"3"
"Muskazine";"977000000026";"3"
"Pineapple bun";"977000000029";"3"
"Schneeball";"977000000033";"3"
"Schuxen";"977000000034";"3"
"Tortell";"977000000041";"3"
```

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.

- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.14.6. Daten an TCP/IP-Port senden

Diese Aktion sendet die Daten an ein externes Gerät, das eine TCP/IP-Verbindung an einer voreingestellten Portnummer akzeptiert.

Daten an TCP/IP-Port senden stellt die Verbindung zum Gerät her, sendet die Daten und beendet dann die Verbindung. Die Verbindung und Kommunikation werden durch den Client-Server-Handshake gesteuert, der bei der Initiierung oder Beendigung der TCP-Verbindung erfolgt.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.

- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Verbindungseinstellungen** legt Verbindungseigenschaften fest.

- **Absender antworten**: Ermöglicht eine direkte Antwort an das Socket, von dem die Triggerdaten stammen. Verwenden Sie diese Option, um Feedback zum Druckprozess zu geben.

  > **ANMERKUNG**
  > Diese Option ist in NiceLabel Automation verfügbar.

  Voraussetzungen für die Einstellung **Absender antworten** sind:

  - Die entfernte Partei schließt den Kommunikationskanal nicht, nachdem die Nachricht gesendet wurde.
  - Die Aktion **Daten an TCP/IP-Port senden** wird innerhalb des **TCP/IP-Server**-Triggers verwendet.
  - Konfigurieren Sie das Ausführungsereignis im **TCP/IP-Servertrigger nicht** als **Bei Trennung des Clients**.

- **Ziel (IP-Adresse:Port)**: Zieladresse und Port des TCP/IP-Servers. Sie können die Verbindungsparameter fest codieren und feste Hostnamen und IP-Adressen verwenden oder variable Verbindungsparameter nutzen, indem Sie auf den Rechtspfeil klicken und eine vordefinierte Variable auswählen. Weitere Informationen finden Sie im Abschnitt Werte in einem Objekt kombinieren des NiceLabel Automation Benutzerhandbuchs.

  Beispiel

  Wenn die Variable **`hostname`** den Namen des TCP/IP-Servers und die Variable **`port`** die Portnummer bereitstellt, geben Sie folgende Parameter für das Ziel ein:

  ```
  [hostname]:[port]
  ```

- **Trennungsverzögerung**: verlängert die Verbindung zum Zielsocket um die eingestellte Dauer, nachdem die Daten übermittelt wurden. Bestimmte Geräte brauchen mehr Zeit, um die Daten zu verarbeiten. Geben Sie den Verzögerungswert manuell ein oder klicken Sie auf die Pfeile, um ihn zu vergrößern oder zu verkleinern.
- **Datenantwort in einer Variablen speichern**: Wählt eine Variable aus (oder erstellt eine Variable), um die Serverantwort zu speichern. Daten, die nach Verstreichen der „Trennungsverzögerung“ vom TCP/IP-Server empfangen werden, werden in dieser Variablen gespeichert.

Die Gruppe **Inhalt** legt den an den TCP/IP-Server zu sendenden Inhalt fest.

TIPP

Sie können festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzugeben, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie eine Variable aus der Liste ein. Weitere Informationen finden Sie im Abschnitt „Werte in einem Objekt kombinieren“ des NiceLabel Automation Benutzerhandbuchs.

- **Daten**: Inhalt, der gesendet werden soll.
- **Codieren**: Codierungstyp für die gesendeten Daten. Auto legt die Codierung automatisch fest. Wählen Sie, falls nötig, den bevorzugten Codierungstyp aus der Dropdown-Liste aus.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ### ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.14.7. Daten an serielle Schnittstelle senden

Diese Aktion sendet Daten an eine serielle Schnittstelle. Verwenden Sie sie, um mit externen Geräten an der seriellen Schnittstelle zu kommunizieren.

TIPP

Stellen Sie sicher, dass die Schnittstelleneinstellungen an beiden Enden übereinstimmen, also in der konfigurierten Aktion und am Gerät an der seriellen Schnittstelle. Die serielle Schnittstelle kann nur von einer einzigen Anwendung auf dem Rechner verwendet werden. Um die Schnittstelle aus dieser Aktion erfolgreich verwenden zu können, darf keine andere Anwendung die Schnittstelle gleichzeitig nutzen, nicht einmal ein Druckertreiber.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Schnittstelle** legt die serielle Schnittstelle fest.

- **Portname**: Name der Schnittstelle, mit der sich das externe Gerät verbindet. Dabei kann es sich entweder um eine Hardware-COM-Schnittstelle oder um eine virtuelle COM-Schnittstelle handeln.

Die Gruppe **Schnittstellen Einstellungen** legt weitere Einstellungen für die Verbindung mit der Schnittstelle fest.

- **Bits pro Sekunde**: Geschwindigkeit, die das externe Gerät für die Kommunikation mit dem PC nutzt. Eine weitere Bezeichnung für diese Einstellung ist „Baudrate“. Wählen Sie den Wert aus dem Dropdown-Menü aus.
- **Datenbits**: Anzahl von Datenbits pro Zeichen. In neueren Geräten werden fast durchgehend 8 Datenbits verwendet. Wählen Sie den Wert aus dem Dropdown-Menü aus.
- **Parität**: Methode der Fehlererkennung bei der Übertragung. Die häufigste Einstellung für die Parität ist „keine“; in diesem Fall wird die Fehlererkennung durch ein Kommunikationsprotokoll gehandhabt (Flusssteuerung). Wählen Sie den Wert aus dem Dropdown-Menü aus.
- **Stoppbits**: stoppt die Bits, die am Ende jedes Zeichens gesendet werden, um der empfangenden Hardware die Erkennung des Endes eines Zeichens sowie die erneute Synchronisierung mit dem

Zeichenstrom zu ermöglichen. Elektronische Geräte verwenden üblicherweise ein einzelnes Stoppbit. Wählen Sie den Wert aus dem Dropdown-Menü aus.

- **Flusssteuerung**: Die serielle Schnittstelle kann Schnittstellensignale nutzen, um die Datenübertragung anzuhalten und fortzusetzen.

Die Gruppe **Inhalt** legt die an die serielle Schnittstelle zu sendenden Inhalte fest.

TIPP

Sie können festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzugeben, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie eine Variable aus der Liste ein. Weitere Informationen finden Sie im Abschnitt „Werte in einem Objekt kombinieren“ des NiceLabel Automation Benutzerhandbuchs.

- **Daten**: Inhalt, der gesendet werden soll.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.14.8. Daten von serieller Schnittstelle lesen

Diese Aktion sammelt Daten, die über die serielle Schnittstelle (RS-232) abgerufen wurden, und speichert sie in einer ausgewählten Variablen. Verwenden Sie diese Aktion, um mit externen Geräten an der seriellen Schnittstelle zu kommunizieren.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Schnittstelle** legt die serielle Schnittstelle fest.

- **Portname**: Name der Schnittstelle, mit der sich ein externes Gerät verbindet. Dabei kann es sich entweder um eine Hardware-COM-Schnittstelle oder um eine virtuelle COM-Schnittstelle handeln.

Die Gruppe **Schnittstellen Einstellungen** legt weitere Einstellungen für die Verbindung mit der Schnittstelle fest.

- **Bits pro Sekunde**: Geschwindigkeit, die das externe Gerät für die Kommunikation mit dem PC nutzt. Eine weitere Bezeichnung für diese Einstellung ist „Baudrate“.
- **Datenbits**: gibt die Anzahl von Datenbits pro Zeichen an. In neueren Geräten werden fast durchgehend 8 Datenbits verwendet.
- **Parität**: gibt die Methode der Fehlererkennung bei der Übertragung an. Die häufigste Einstellung für die Parität ist „keine“; in diesem Fall wird die Fehlererkennung durch ein Kommunikationsprotokoll gehandhabt (Flusssteuerung).
- **Stoppbits**: stoppt die Bits, die am Ende jedes Zeichens gesendet werden, um der empfangenden Hardware die Erkennung des Endes eines Zeichens sowie die erneute Synchronisierung mit dem

Zeichenstrom zu ermöglichen. Elektronische Geräte verwenden üblicherweise ein einzelnes Stoppbit.

- **Flusssteuerung**: Die serielle Schnittstelle kann Schnittstellensignale nutzen, um die Datenübertragung anzuhalten und fortzusetzen.

Beispiel

Ein langsames Gerät muss eventuell einen Handshake mit der seriellen Schnittstelle ausführen, um anzuzeigen, dass die Übertragung angehalten werden muss, während das Gerät die bereits empfangenen Daten verarbeitet.

Die Gruppe **Optionen** enthält die folgenden Einstellungen:

- **Leseverzögerung**: optionale Verzögerung beim Lesen der Daten von der seriellen Schnittstelle. Nach der Verzögerung wird der gesamte Inhalt des Puffers der seriellen Schnittstelle gelesen. Geben Sie die Verzögerung manuell ein oder klicken Sie auf die Pfeile, um den Wert zu vergrößern oder zu verkleinern.
- **Initialisierungsdaten senden**: gibt die Zeichenfolge an, die an die ausgewählte serielle Schnittstelle gesendet wird, bevor die Daten gelesen werden. So kann die Aktion das jeweilige Gerät für die Bereitstellung der Daten initialisieren. Die Option kann auch verwendet werden, um eine spezifische Frage an das Gerät zu senden und eine spezifische Antwort zu erhalten. Klicken Sie auf die Pfeiltaste, um Sonderzeichen einzugeben.

Die Gruppe **Datenextraktion** legt fest, wie die definierten Teile der empfangenen Daten extrahiert werden.

- **Startposition**: Startposition für die Datenextraktion.
- **Endposition**: Endposition für die Datenextraktion.

Die Gruppe **Ergebnis** definiert eine Variable zum Speichern von Dateien.

- **Daten in Variable speichern**: Wählen Sie eine Variable aus (oder erstellen Sie eine Variable), um die empfangenen Daten zu speichern.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.14.9. Daten an Drucker senden

Diese Aktion sendet Daten an einen ausgewählten Drucker. Verwenden Sie sie, um zuvor erzeugte Druckerströme an einen verfügbaren Drucker zu senden.

Das Modul NiceLabel Automation nutzt den installierten Druckertreiber im Pass-Through-Modus, um Daten an die Zielschnittstelle senden zu können, mit der der Drucker verbunden ist, z. B. LPT, COM, TCP/IP oder USB.

> **ANMERKUNG**
> Mögliches Szenario. Vom Trigger empfangene Daten müssen auf demselben Netzwerkdrucker, aber auf unterschiedlichen Etikettenvorlagen (.NLBL-Etikettendateien) gedruckt werden. Der Drucker kann Daten von verschiedenen Rechnern entgegennehmen und druckt Aufträge für gewöhnlich in der Reihenfolge des Empfangs. Das Modul Automation Builder sendet jede Etikettenvorlage in einem separaten Druckauftrag, sodass ein anderer Rechner seinen Druckauftrag zwischen den in unserem Automation Builder Modul erstellten Druckaufträgen einfügen kann. Anstatt jeden Auftrag separat an den Drucker zu senden, können Sie alle Etikettenaufträge verbinden (durch Nutzung der Aktion Druck an Datei umleiten) und dann als einzelnen „großen“ Druckauftrag an den Drucker senden.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Drucker** wählt den Drucker aus.

- **Druckername**: Name des Druckers, an den die Daten gesendet werden sollen. Wählen Sie den Drucker aus der Dropdown-Liste lokal installierter Druckertreiber aus und geben Sie einen benutzerdefinierten Druckernamen ein oder definieren Sie ihn dynamisch anhand einer vorhandenen oder neu erstellten Variablen.

Die Gruppe **Datenquelle** legt den Inhalt fest, der an den Drucker gesendet werden soll.

- **Vom Trigger empfangene Daten verwenden:** vom Trigger empfangene Daten werden verwendet. In diesem Fall wollen Sie den empfangenen Druckdatenstrom als Eingabe für den Filter verwenden. Ihr Ziel besteht darin, ihn ohne Änderungen an einen Drucker umzuleiten. Dasselbe Ergebnis kann erzielt werden, indem Sie die interne Variable **`DataFileName`** aktivieren und die Inhalte der Datei verwenden, auf die sie verweist. Weitere Informationen finden Sie im Abschnitt „Zusammengesetzte Werte verwenden“ im NiceLabel Automation Benutzerhandbuch.
- **Dateiname**: Pfad und Name der Datei, die einen Druckdatenstrom enthält. Der Inhalt der angegebenen Datei wird an einen Drucker gesendet. Wählen Sie **Datenquelle**, um den Dateinamen dynamisch anhand eines Variablenwerts zu definieren.
- **Variable**: Variable (vorhanden oder neu), die den Druckdatenstrom enthält.
- **Benutzerdefiniert**: legt den benutzerdefinierten Inhalt fest, der an einen Drucker gesendet werden soll. Sie können festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzugeben, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie eine Variable aus der Liste ein. Weitere Informationen finden Sie im Abschnitt „Werte in einem Objekt kombinieren“ des NiceLabel 10 Benutzerhandbuchs.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.

- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.14.10. HTTP-Anfrage

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **LMS Enterprise**.

Diese Aktion sendet anhand der ausgewählten HTTP-Methode Daten an den Ziel-Webserver. Die Schemata HTTP und HTTPS sind erlaubt.

HTTP fungiert als Anfrage-Antwort-Protokoll im Client-Server-Modell. In dieser Aktion fungiert NiceLabel 10 als Client, der mit einem Remote-Server kommuniziert. Die Aktion übermittelt eine ausgewählte HTTP-Anfragenachricht an einen Server. Der Server gibt daraufhin eine Antwortnachricht aus, deren Textkörper Statusinformationen zur Ausführung der Anfrage sowie angeforderte Inhalte enthalten kann.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Verbindungseinstellungen** legt Verbindungsparameter fest.

ANMERKUNG

Diese Aktion unterstützt Internet Protocol Version 6 (IPv6).

- **Ziel**: Adresse, Port und Ziel (Pfad) auf dem Webserver.

TIPP

Wenn ein Webserver am Standardport 80 ausgeführt wird, können Sie die Portnummer auslassen. Sie sollten die Verbindungsparameter fest codieren und einen festen Hostnamen oder eine feste IP-Adresse verwenden. Verwenden Sie einen Variablenwert, um diese Option dynamisch festzulegen. Weitere Informationen finden Sie im Abschnitt „Zusammengesetzte Werte verwenden“ im NiceLabel Automation Benutzerhandbuch.

Beispiel

Wenn die Variable **`hostname`** den Namen des Webservers und die Variable **`port`** die Portnummer bereitstellt, können Sie Folgendes für das Ziel eingeben:

```
[hostname]:[port]
```

- **Anfragemethode**: verfügbare Anfragemethoden.
- **Zeitüberschreitung**: Zeitüberschreitungsdauer (in ms), innerhalb derer die Serververbindung hergestellt und eine Antwort empfangen werden sollte.
- **Statusantwort in einer Variable speichern**: Variable, die den vom Server empfangenen Statuscode speichert.

TIPP

Statuscodes im Bereich 2XX geben an, dass der Vorgang erfolgreich war. Die normale OK-Antwort ist Code 200. Die Codes im Bereich „5XX“ zeigen Serverfehler an.

- **Datenantwort in einer Variable speichern**: Variable, die die vom Server empfangenen Daten speichert.

Die Gruppe **Authentifizierung** ermöglicht es Ihnen, die Verbindung zum Webserver zu sichern.

- **Grundlegende Authentifizierung aktivieren**: ermöglicht es Ihnen, die erforderlichen Zugangsdaten zur Verbindung mit dem Webserver einzugeben. Benutzername und Passwort können entweder fest codiert sein oder anhand einer Variablen angegeben werden.

ANMERKUNG

Die grundlegende HTTP-Authentifizierung (BA) nutzt statische Standard-HTTP-Header. Der BA-Mechanismus bietet keinen Datenschutz für die übermittelten Zugangsdaten. Sie werden bei der Übertragung lediglich mit Base64 codiert, aber nicht verschlüsselt. Die grundlegende Authentifizierung sollte bei Übertragungen per HTTPS genutzt werden.

- **Passwort anzeigen**: blendet das Passwort ein.

Die Gruppe **Inhalt** legt die an den Webserver zu sendenden Inhalte fest.

- **Daten**: Inhalt, der gesendet werden soll. Sie können festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzugeben, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie die Variable aus der Liste ein. Weitere Informationen finden Sie im Abschnitt „Werte in einem Objekt kombinieren“ des NiceLabel 10 Benutzerhandbuchs.
- **Codieren**: Codierungstyp für die gesendeten Daten.

  ### TIPP

  Auto legt die Codierung automatisch fest. Wählen Sie, falls nötig, den bevorzugten Codierungstyp aus der Dropdown-Liste aus.

- **Typ**: Content-Type-Eigenschaft der HTTP-Nachricht. Wird keine Auswahl getroffen, wird der Standard **`application/x-www-form-urlencoded`** verwendet. Wenn der entsprechende Typ nicht aufgeführt ist, legen Sie einen benutzerdefinierten Typ fest oder stellen Sie eine Variable ein, die den Typ dynamisch definiert.

Zusätzliche HTTP-Header werden von bestimmten HTTP-Servern angefordert (insbesondere für REST-Dienste).

- **Zusätzliche Header**: fest codierte Header oder Header, die aus Variablenwerten bezogen werden. Um auf die Variablen zuzugreifen, klicken Sie auf die kleine Pfeil-Schaltfläche rechts neben dem Textbereich. Weitere Informationen finden Sie im Abschnitt „Werte in einem Objekt kombinieren“ des NiceLabel 10 Benutzerhandbuchs.
  Bestimmte HTTP-Server erfordern (insbesondere für REST-Dienste) die Aufnahme benutzerdefinierter HTTP-Header in die Nachricht. In diesem Abschnitt erfahren Sie, wie Sie die erforderlichen HTTP-Header angeben können.

Die HTTP-Header müssen anhand der folgenden Syntax eingegeben werden:

```
header field name: header field value
```

Um beispielsweise die Header-Feldnamen **`Accept`**, **`User-Agent`** und **`Content-Type`** zu nutzen, könnten Sie folgende Syntax verwenden:

```
Accept: application/json; charset=utf-8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/31.0.1650.63 Safari/537.36
Content-Type: application/json; charset=UTF-8
```

Sie können die Header-Feldnamen fest codieren oder ihre Werte aus Trigger-Variablen beziehen. Verwenden Sie so viele benutzerdefinierte Header-Felder wie nötig, aber achten Sie daran, dass jedes Header-Feld in einer neuen Zeile platziert wird.

ANMERKUNG

Die eingegeben HTTP-Header heben die Header auf, die bereits anderswo in den Aktionseigenschaften definiert sind, z. B. **Content-Type**.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (**`wahr`** oder **`falsch`**). Ist das Ergebnis des Ausdrucks **`wahr`**, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.14.11. Webdienst

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **LMS Enterprise**.

Webdienst ist eine Methode der Kommunikation zwischen zwei elektronischen Geräten oder zwischen Softwareinstanzen. Webdienst ist als Standard für den Datenaustausch definiert. Er nutzt das XML-Format zum Markieren der Daten, das SOAP-Protokoll zum Übertragen der Daten und die WSDL-Sprache zur Beschreibung der verfügbaren Dienste.

Diese Aktion verbindet sich mit einem entfernten Webdienst und wendet die Methoden auf ihm an. Methoden können als Aktionen beschrieben werden, die auf dem Webdienst veröffentlicht sind. Diese Aktion sendet eingehende Werte an die ausgewählte Methode im entfernten Webdienst, ruft das Ergebnis ab und speichert es in ausgewählten Variablen.

Nachdem Sie die WSDL importiert und eine Referenz auf den Webdienst hinzugefügt haben, werden dessen Methoden im Kombinationsfeld Methode aufgelistet.

ANMERKUNG

Sie können einfache Datentypen über den Webdienst übertragen, z. B. String, Integer und Boolean, aber keine komplexen Datentypen. Die WSDL darf nur eine Bindung enthalten.

ANMERKUNG

Sie möchten Produktetiketten drucken. Ihr Trigger würde nur einen Teil der erforderlichen Daten erhalten. Zum Beispiel: Der Trigger empfängt den Wert für die Variablen **`Product ID`** und **`Description`**, aber nicht für **`Price`**. Die Preisinformationen befinden sich in einer separaten Datenbank, auf die anhand eines Webdienst-Aufrufs zugegriffen werden kann. Webdienst definiert die Funktion anhand einer WSDL-Definition. Beispielsweise ist die Funktionseingabe **`Product ID`** und die Ausgabe ist **`Price`**. Die Webdienst-Aktion sendet **`Product ID`** an den Webdienst. Sie wird ausgeführt, führt eine interne Suche in ihrer Datenbank aus und stellt den entsprechenden **`Price`** als Ergebnis bereit. Die Aktion speichert das Ergebnis in einer Variablen, die auf dem Etikett verwendet werden kann.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Webdienstdefinition** enthält die folgenden Einstellungen:

ANMERKUNG

Diese Aktion unterstützt Internet Protocol Version 6 (IPv6).

- **WSDL**: Speicherort der WSDL-Definition.
  Die WSDL wird normalerweise vom Webdienst bereitgestellt. Normalerweise geben Sie den Link zur WSDL ein und klicken auf **Importieren**, um die Definition zu lesen. Falls Sie Probleme beim Abrufen der WSDL von der Online-Ressource haben, speichern Sie die WSDL als Datei und geben Sie den Pfad mit dem Dateinamen an, um die Methoden zu laden.NiceLabel 10 erkennt automatisch, ob der entfernte Webdienst eine Dokument- oder RPC-Syntax nutzt, und ob er entsprechend kommuniziert oder nicht.
- **Adresse**: Adresse, unter der der Webdienst veröffentlicht ist.
  Zu Beginn wird diese Information vom WSDL abgerufen, kann aber vor Ausführung der Aktion aktualisiert werden. Dies ist hilfreich für geteilte Entwicklungs-/Test-/Produktionsumgebungen, in denen dieselbe Liste von Aktionen, aber andere Namen von Servern, die Webdienste ausführen, verwendet werden.
  Sie können festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzugeben, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie die Variable aus der Liste ein. Weitere Informationen finden Sie im Abschnitt „Werte in einem Objekt kombinieren“ des NiceLabel 10 Benutzerhandbuchs.

- **Methode**: Methoden (Funktionen), die für den ausgewählten Webdienst verfügbar sind. Die Liste wird von der WSDL-Definition automatisch ausgefüllt.
- **Parameter**: Eingabe- und Ausgabevariablen der ausgewählten Methode (Funktion).
  Eingabeparameter erwarten eine Eingabe. Zu Test- und Fehlerbehebungszwecken können Sie einen festen Wert eingeben und eine Vorschau des Ergebnisses anzeigen. Normalerweise würden Sie jedoch eine Variable für eingehende Parameter auswählen. Der Wert dieser Variablen wird als Eingabeparameter verwendet. Der ausgehende Parameter stellt die Ergebnisse der Funktion bereit. Sie müssen die Variable auswählen, die das Ergebnis speichern soll.
- **Zeitüberschreitung**: Zeitüberschreitung, nach der die Verbindung mit einem Server hergestellt wird.

Authentifizierung aktiviert eine einfache Benutzerauthentifizierung. Die Option gibt Benutzerdaten an, die zur Herstellung eines ausgehenden Aufrufs eines entfernten Webdienstes benötigt werden.

- **Grundlegende Authentifizierung aktivieren:** ermöglicht die Angabe des **Benutzernamens** und **Passworts**, die manuell eingegeben oder durch Variablenwerte definiert werden können. Wählen Sie **Datenquellen**, um die Variablen auszuwählen oder zu erstellen.
- **Passwort anzeigen:** zeigt die verdeckten Zeichen im **Benutzernamen** und im **Passwort** an.
  Details zu Sicherheitsfragen finden Sie im Abschnitt „Zugriff auf Ihre Trigger sichern“ des NiceLabel Automation Benutzerhandbuchs.

Das Feld **Datenvorschau** ermöglicht es Ihnen, den Webdienst testweise auszuführen.

- Die Schaltfläche **Ausführen** führt einen Webdienst-Aufruf aus.
  Sendet Werte eingehender Parameter an den Webdienst und stellt das Ergebnis im ausgehenden Parameter bereit. Verwenden Sie diese Funktion, um die Ausführung eines Webdienstes zu testen. Sie können Werte für eingehende Parameter eingeben und das Ergebnis auf dem Bildschirm anzeigen. Wenn Sie mit der Ausführung zufrieden sind, können Sie den eingegebenen Festwert für den eingehenden Parameter durch eine Variable aus der Liste ersetzen.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.15. Sonstiges

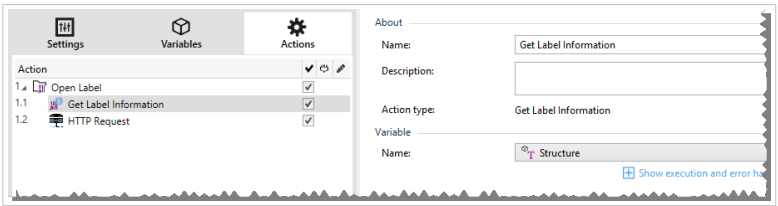
5.15.1. Etiketteninformationen abrufen

> **PRODUKTEBENEN-INFO**
> Automation Builder-Funktionen erfordern **LMS Enterprise**.

Diese Aktion gibt Strukturinformationen über die verbundene Etikettendatei aus. Sie bietet Informationen zu den Etikettenabmessungen und dem Druckertreiber sowie eine Liste aller Etikettenvariablen und ihrer Haupteigenschaften.

Die Aktion „Etiketteninformationen abrufen“ gibt die Originalinformationen aus, wie sie in der Etikettendatei gespeichert sind. Außerdem bietet sie Informationen nach Simulation des Druckprozesses. Die Simulation stellt sicher, dass alle Etikettenvariablen denselben Wert wie bei einem normalen Druckvorgang erhalten. Auch bei den Angaben zur Höhe des Etiketts handelt es sich um die tatsächlichen Abmessungen, falls Sie das Etikett als Etikett mit variabler Höhe definiert haben (in diesem Fall hängt die Etikettengröße von der zu druckenden Datenmenge ab). Die Aktion gibt die Abmessungen für die Etikettengröße aus, nicht für die Seitengröße.

Die Aktion speichert Strukturinformationen zum Etikett in einer ausgewählten Variablen. Sie können die Daten anhand der Aktion „HTTP-Anfrage“ (oder einer ähnlichen Aktion für ausgehende Datenverbindungen) oder aber in der Trigger-Antwort (bei Verwendung eines bidirektionalen Triggers) zurück an das System senden.



ANMERKUNG

Diese Aktion muss unter der Aktion Etikett öffnen eingebunden werden.

Die Gruppe **Variable** wählt eine Variable aus (oder erstellt sie), in der die strukturellen Informationen zu einem Etikett gespeichert werden.

- **Name:** gibt den Variablennamen an. Wählen Sie eine Variable aus (oder erstellen Sie eine Variable), die die Etiketteninformationen im XML-Format speichert.
    - Wenn Sie die Informationen aus der XML-Datei in diesem Trigger nutzen möchten, können Sie sie mithilfe der Aktion „Datenfilter verwenden“ definieren und ausführen (nur Automation Builder).
    - Wenn Sie die XML-Daten als Antwort in Ihren HTTP- oder Webdienst-Trigger ausgeben möchten, verwenden Sie diese Variable direkt im Feld **Antwortdaten** der Trigger-Konfigurationsseite.
    - Wenn Sie die XML-Daten als Datei speichern möchten, verwenden Sie die Aktion Daten in Datei speichern.

Die Gruppe **Zusätzliche Einstellungen** ermöglicht es Ihnen, die Nutzung vorläufiger Werte zu aktivieren.

- **Vorläufige Werte verwenden:** ersetzt fehlende Datenquellenwerte durch vorläufige Werte.

TIPP

Im Abschnitt „Variable“ des NiceLabel 10 Designer Benutzerhandbuchs finden Sie eine detaillierte Beschreibung von vorläufigen Werten.

Beispiel für Etiketteninformationen im XML-Format

Das folgende Beispiel zeigt eine Strukturansicht der Etikettenelemente und ihrer Attribute wie ausgegeben.

```
<?xml version="1.0" encoding="UTF-8"?>
<Label>
```

```
    <Original>
        <Width>25000</Width>
        <Height>179670</Height>
        <PrinterName>QLS 3001 Xe</Printer>
    </Original>
    <Current>
        <Width>25000</Width>
        <Height>15120</Height>
        <PrinterName>QLS 3001 Xe</Printer>
    </Current>
    <Variables>
        <Variable>
            <Name>barcode</Name>
            <Description></Description>
            <DefaultValue></DefaultValue>
            <Format>All</Format>
            <CurrentValue></CurrentValue>
            <IncrementType>None</IncrementType>
            <IncrementStep>0</IncrementStep>
            <IncrementCount>0</IncrementCount>
            <Length>100</Length>
        </Variable>
    </Variables>
</Format>
```

XML-Spezifikation für Etiketteninformationen

Dieser Abschnitt enthält eine Beschreibung der XML-Dateistruktur, die von der Aktion „Etiketteninformationen abrufen“ ausgegeben wird.

ANMERKUNG

Alle Abmessungswerte sind als 1/1000 mm angegeben. Eine Breite von 25000 steht beispielsweise für 25 mm.

- **<Label>:** dies ist ein Stammelement.
- **<Original>:** gibt Etikettenabmessungen und Druckernamen wie in der Etikettendatei gespeichert an.
  - **Width:** dieses Element enthält die ursprüngliche Breite des Etiketts.
  - **Height:** dieses Element enthält die ursprüngliche Höhe des Etiketts.
  - **PrinterName:** dieses Element enthält den Druckernamen, für den das Etikett erstellt wurde.
- **Current:** gibt Etikettenabmessungen und Druckernamen nach Ausführung des simulierten Druckvorgangs an.

- **Width:** dieses Element enthält die tatsächliche Breite des Etiketts.
- **Height:** dieses Element enthält die tatsächliche Höhe des Etiketts. Wenn ein Etikett als Etikett mit variabler Höhe definiert ist, kann seine Höhe zusammen mit anderen Etikettenobjekten gesteigert werden. Textfelder und RTF-Objekte z. B. können sich vertikal ausdehnen, was auch eine Ausdehnung des gesamten Etiketts bewirkt.
- **PrinterName:** dieses Element enthält den Namen des Druckers, der für den Druck genutzt wird.

Beispiel

Wenn der Treiber des ursprünglichen Druckers nicht auf diesem Computer installiert ist oder der Drucker anhand der Aktion Drucker einstellen geändert wurde, wird ein anderer Drucker verwendet.

- **<Variables> und <Variable>:** das Element `Variables` enthält eine Liste aller Abfragevariablen auf dem Etikett, wobei jede einzelne in einem separaten `Variable`-Element definiert ist. Die Abfragevariablen sind diejenigen Variablen, die beim Drucken des Etiketts aus NiceLabel 10 im Druck-Dialogfeld aufgelistet sind. Gibt es keine Abfragevariablen auf dem Etikett, ist das Element `Variables` leer.
  - **Name:** enthält den Variablennamen.
  - **Beschreibung:** enthält die Beschreibung der Variablen.
  - **DefaultValue:** enthält den Standardwert gemäß Definition für die Variable während des Etikettendesign-Prozesses.
  - **Format:** enthält den akzeptablen Typ von Variableninhalt (Zeichen).
  - **IsPrompted:** enthält Informationen dazu, ob die Variable zum Druckzeitpunkt abgefragt wird oder nicht.
  - **PromptText:** enthält den Text, der Anwender zur Eingabe eines Wertes auffordert.
  - **CurrentValue:** enthält den tatsächlichen, für den Druck verwendeten Wert.
  - **IncrementType:** enthält Informationen dazu, ob die Variable als Zähler festgelegt ist oder nicht. Wenn sie als Zähler festgelegt ist, wird außerdem die Art von Zähler angegeben.
  - **IncrementStep:** enthält Informationen über den Zählerschritt. Der Zählerwert nimmt gemäß diesem Wert beim nächsten Etikett zu/ab.
  - **IncrementCount:** enthält Informationen darüber, wann der Zählerwert zu-/abnehmen soll. Für gewöhnlich ändert sich der Zählerwert bei jedem Etikett, aber dies kann geändert werden.
  - **Length:** enthält die maximale Anzahl von in einer Variable gespeicherten Zeichen.
  - **IsPickListEnabled:** enthält Informationen dazu, ob der Benutzer Variablenwerte aus einer Auswahlliste auswählt oder nicht.
  - **PickListValues:** enthält die tatsächlichen (auswählbaren) Auswahllistenwerte.

XML-Schema (XSD) für Etikettenspezifikationen im XML-Format

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Format" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Label">
        <xs:complexType>
            <xs:all>
                <xs:element name="Original">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Width" type="xs:decimal"
minOccurs="1" />
                            <xs:element name="Height" type="xs:decimal"
minOccurs="1" />
                            <xs:element name="PrinterName" type="xs:string"
minOccurs="1" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Current">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Width" type="xs:decimal"
minOccurs="1" />
                            <xs:element name="Height" type="xs:decimal"
minOccurs="1" />
                            <xs:element name="PrinterName" type="xs:string"
minOccurs="1" />
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Variables">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Variable" minOccurs="0"
maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="Name"
type="xs:string" minOccurs="1" />
                                        <xs:element name="Description"
type="xs:string" minOccurs="1" />
                                        <xs:element name="DefaultValue"
type="xs:string" minOccurs="1" />
                                        <xs:element name="Format"
type="xs:string" minOccurs="1" />
                                        <xs:element name="CurrentValue"
```

```
type="xs:string" minOccurs="1" />
                                        <xs:element name="IncrementType"
type="xs:string" minOccurs="1" />
                                        <xs:element name="IncrementStep"
type="xs:integer" minOccurs="1" />
                                        <xs:element name="IncrementCount"
type="xs:integer" minOccurs="1" />
                                        <xs:element name="Length"
type="xs:string" minOccurs="1" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:all>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

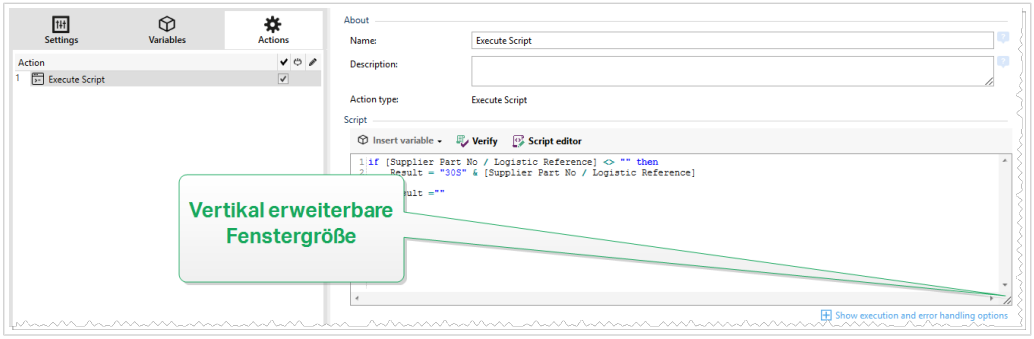
- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.15.2. Script ausführen

Diese Aktion erweitert den Funktionsumfang der Software durch Verwendung benutzerdefinierter VBScript- oder Python-Skripte. Verwenden Sie diese Funktion, falls die integrierten Aktionen Ihren Datenbearbeitungsanforderungen nicht entsprechen.

Skripte können die Trigger-Variablen enthalten – dabei kann es sich sowohl um interne Variablen als auch um Variablen handeln, die definiert oder aus Etiketten importiert werden.

Stellen Sie sicher, dass das Windows-Konto, unter dem der Dienst ausgeführt wird, über die nötigen Berechtigungen zur Ausführung der Befehle im Skript verfügt.



> **ANMERKUNG**
> Der Skript-Typ wird nach Trigger in den Trigger-Eigenschaften konfiguriert. Alle „Script ausführen“-Aktionen innerhalb eines einzigen Triggers müssen denselben Typ aufweisen.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.

- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Der **Skript**-Editor bietet die folgenden Funktionen:

- **Datenquelle einfügen**: fügt eine vorhandene oder neu erstellte Variable in das Skript ein.
- **Überprüfen:** validiert die eingegebene Skript-Syntax.
- **Skript-Editor:** öffnet den Editor, der Scripting einfacher und effizienter macht.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

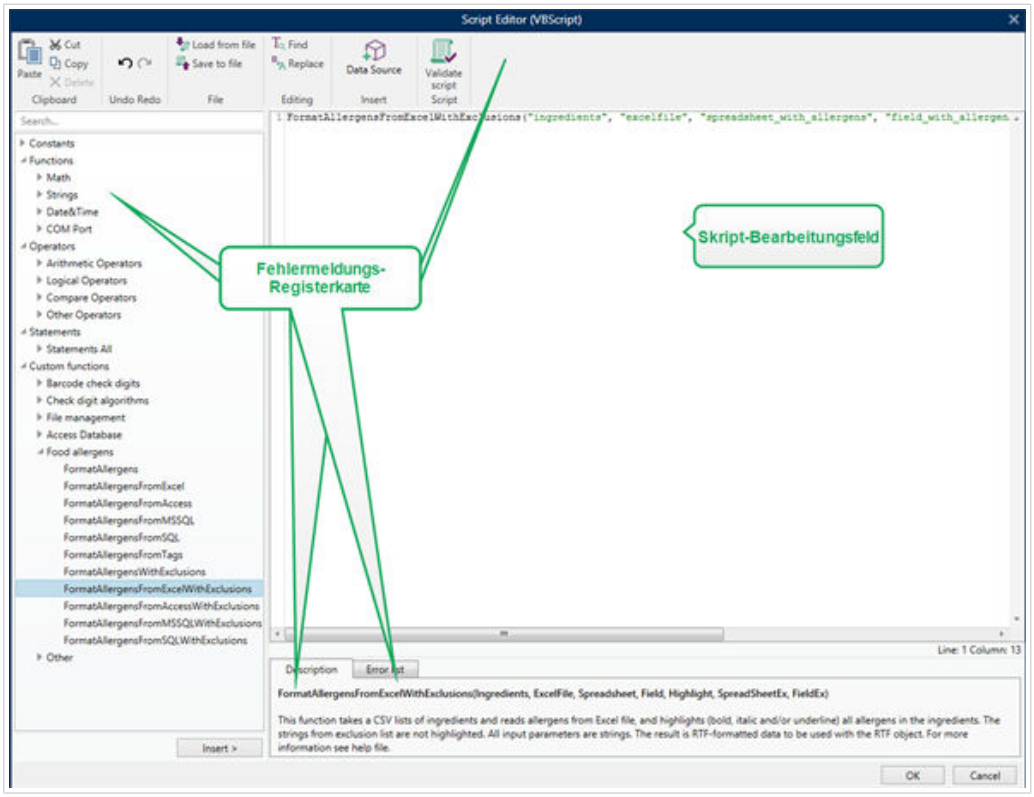
Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.15.2.1. Skript-Editor

NiceLabel 10 bietet einen Skript-Editor, der Ihre Scripting-Aktivitäten mit Python oder VBScript einfacher, fehlerfrei und zeiteffizient macht.



Die Auswahl der Scripting-Sprachen, die im Skript-Editor verwendet werden sollten, variiert zwischen NiceLabel Designer Pro und Automation Builder:

- In Designer müssen Sie auf die Maskendesign-Oberfläche doppelklicken, um Maskeneigenschaften > Zusätzliche Einstellungen > Scripting-Sprache für die Maske zu öffnen.
- In Automation Builder gehen Sie auf Konfigurationselemente > Klick auf Bearbeiten, um die Trigger-Eigenschaften zu öffnen > Einstellungen > Andere > Scripting.

NiceLabel 10 nutzt eine .NET-Variante von Python namens IronPython. Sie fungiert als vollständig kompatible Implementierung der Python-Skriptsprache, welche auch .NET-Methoden unterstützt.

Die Editor-Multifunktionsleiste enthält häufig verwendete Befehle, die auf mehrere Funktionsgruppen verteilt sind.

- Die Zwischenablage-Gruppe bietet Ausschneiden-, Kopieren-, Einfügen- und Löschen-Befehle.
- Die Gruppe Rückgängig und Rückgängig aufheben ermöglicht es, Scripting-Aktionen rückgängig zu machen oder zu wiederholen.
- Die Datei-Gruppe ermöglicht das Laden und Speichern von Skripten in einer Datei.
    - Aus Datei laden: lädt ein Skript aus einer zuvor gespeicherten Textdatei.

    - Als Datei speichern: speichert das aktuelle Skript in einer Textdatei.
- Die Bearbeiten-Gruppe ermöglicht es, Zeichenfolgen in einem Skript zu finden und zu ersetzen.
    - Suchen: sucht die eingegebene Zeichenfolge im Skript.
    - Ersetzen: ersetzt die Zeichenfolge im Skript.
- Einfügen-Gruppe: Der Befehl Datenquelle fügt vorhandene oder neu definierte Datenquellen in das Skript ein.
- Skript-Gruppe: Der Befehl Skript prüfen überprüft die Syntax des eingegebenen Skripts.

Verfügbare Scripting-Elemente enthält alle verfügbaren Elemente, die beim Erstellen eines Skripts verwendet werden können. Durch Doppelklicken auf das Element oder Klicken auf die Einfügen-Schaltfläche wird das Element an der Cursorposition in das Skript eingefügt.

Die Elementbeschreibung bietet grundlegende Informationen über das eingefügte Skript-Element.

Die Fehlerliste enthält die Fehler, die nach Ausführung des Befehls Skript prüfen ausgegeben wurden.

5.15.3. Meldung

Verwenden Sie die Aktion „Meldung“ zum Verfassen benutzerdefinierter Zeichenfolgen (zum Beispiel benutzerdefinierte Warnmeldungen, Variablenwerte und Kommentare). Die Aktion „Meldung“ erzeugt in den Protokolldateien in Ihrem Automation Manager benutzerdefinierte Einträge. Automatisierungsprotokolldateien enthalten anwendungsgenerierte Informationen, Warnungen und Fehlerbeschreibungen. Verwenden Sie Meldungsprotokolle zur Nachverfolgung Ihrer Meldungsvariablen währen der Konfiguration, Fehlerbehebung und beim Debugging.

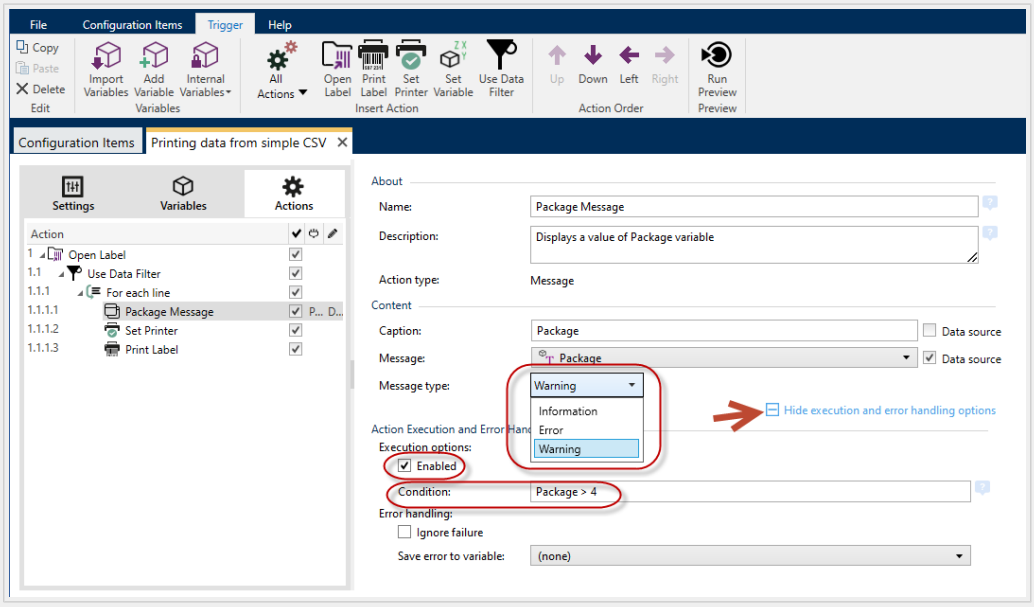
So konfigurieren Sie Meldungsaktionen:

1. Gehen Sie zu **Alle Aktionen** und wählen Sie im Drop-down-Menü **Aktion** die Option **Meldung** aus.
2. Benennen Sie Ihre Aktion um und fügen Sie Ihre Beschreibung ein.
3. Konfigurieren Sie Ihre Meldung **Inhalt**: **Überschrift**, **Meldung**, und **Nachrichtentyp**. **Nachrichtentypen** umfassen:
    - **Information**
    - **Fehler**
    - **Warnung**
4. Erweitern Sie **Optionen zur Ausführung und Fehlerhandhabung anzeigen**, um die Bedingungen zum Anzeigen von Meldungen, Ignorieren von Fehlern und Speichern von Automatisierungsfehlern in Variablen festzulegen.

Der NiceLabel Automation Manager zeigt im Bereich Ihres Automatisierungsprotokolls farbige Meldungen an (zum Beispiel Fehler in Rot und Warnungen in Orange).

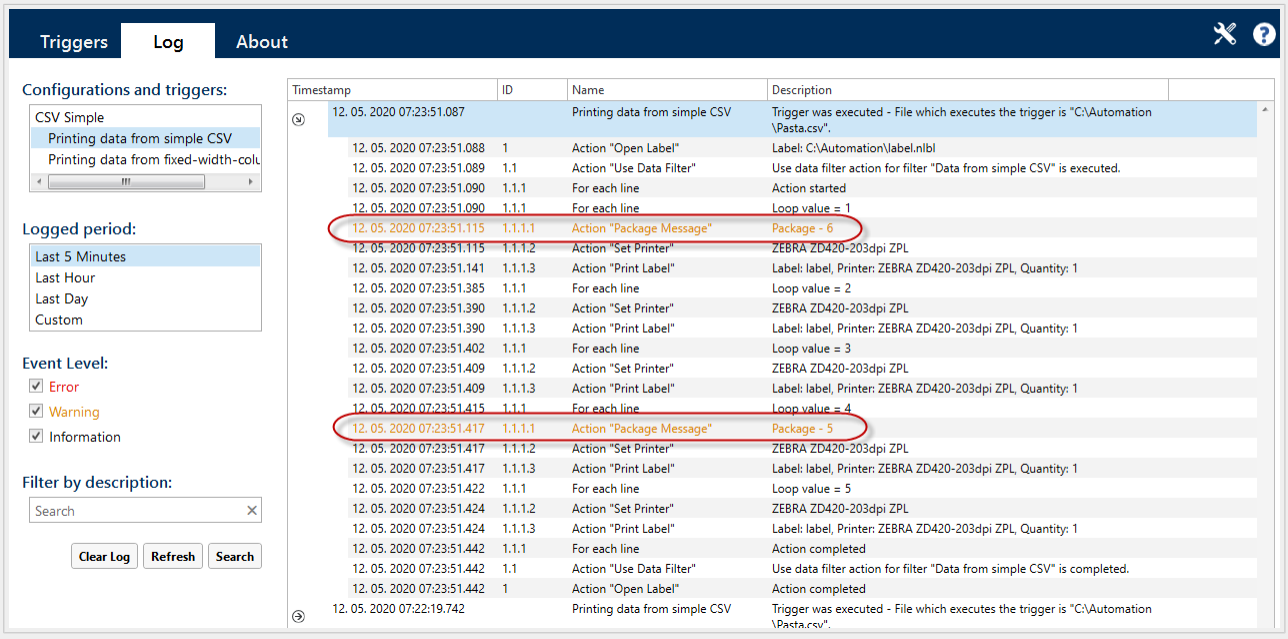
Beispiel:

Sie drucken mit Hilfe der Automatisierung Etiketten für Pasta aus. Ihr Trigger erhält von den ERP-generierten CSV-Dateien Variablenwerte. Wenn Ihr Variablenwert „Verpackung“ größer als 4 ist, erzeugt das Automatisierungsprotokoll eine Warnung.



*Konfigurieren von **Meldung**-Aktionen.*

Im Automation Manager sehen Sie folgendes Ergebnis:



ANMERKUNG

Wenn Sie den Schweregrad Ihrer Meldung auf „Fehler“ festlegen, gehen Ihre Trigger nicht in den Fehlerstatus. Das Drucken ist weiterhin möglich.

Nutzen Sie Meldungsprotokolle für Folgendes:

- Behebung von Konfigurationsfehlern
- Debugging Ihrer Lösungen
- Nachverfolgen der Werte Ihrer gewählten Variablen

Anzeigen Ihrer benutzerdefinierten Warnungen und Fehlermeldungen

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.15.4. Lizenz verifizieren

Diese Aktion liest die aktivierte Lizenz und führt die unter dieser Aktion geschachtelten Aktionen aus, sofern ein bestimmter Lizenztyp verwendet wird.

TIPP

Die Aktion „Lizenz verifizieren“ schützt Ihre Trigger-Konfiguration vor der Ausführung auf nicht autorisierten Rechnern.

ANMERKUNG

Der Lizenzschlüssel, der die Software aktiviert, kann auch die Lösungs-ID codieren. Dies ist eine eindeutige Nummer, die den Lösungsanbieter identifiziert, der die NiceLabel 10-Lizenz verkauft hat.

Wenn die konfigurierte Lösungs-ID der in der Lizenz codierten Lösungs-ID entspricht, kann der Zielrechner geschachtelte Aktionen ausführen; so wird die Ausführung effektiv auf Lizenzen beschränkt, die vom jeweiligen Lösungsanbieter verkauft wurden.

Die Trigger können weiterhin verschlüsselt und gesperrt werden, sodass nur autorisierte Benutzer die Konfiguration öffnen dürfen. Weitere Informationen finden Sie im Abschnitt „Trigger-Konfiguration vor Bearbeitung schützen“ im NiceLabel Automation Benutzerhandbuch.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Lizenzinformation** ermöglicht Ihnen die Auswahl der Lizenz-ID.

- **Lizenz-ID:** definiert die ID-Nummer der Lizenzen, welche die geschachtelten Aktionen ausführen dürfen.
    - Wenn der eingegebene Wert nicht der Lizenz-ID entspricht, die in der Lizenz kodiert ist, werden die geschachtelten Aktionen nicht ausgeführt.
    - Wenn der eingegebene Wert auf 0 gesetzt ist, werden die Aktionen ausgeführt, sofern eine gültige Lizenz gefunden wird.

> **ANMERKUNG**
> Auch die Digital Partner UID kann als Lizenz-ID verwendet werden. Diese Option ist für Mitglieder des NiceLabel Digital Partner Programms verfügbar.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.15.5. Testen

PRODUKTEBENEN-INFO

Automation Builder-Funktionen erfordern **LMS Enterprise**.

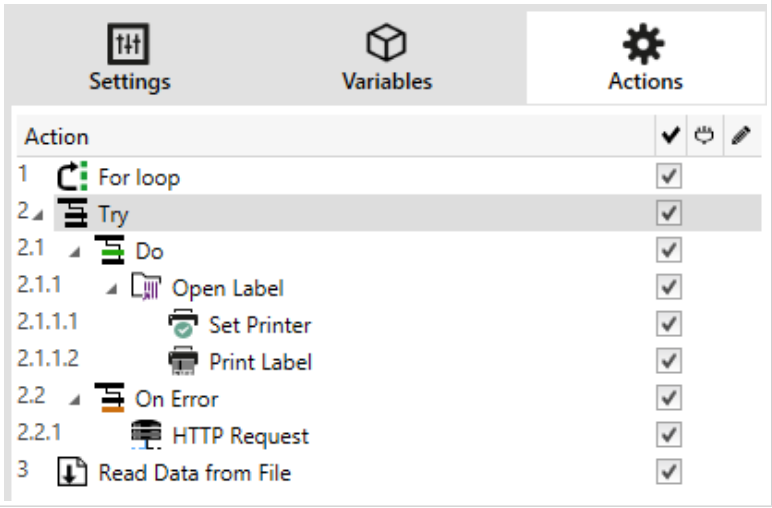
Mit dieser Aktion können Sie:

- Fehler während der Ausführung von Aktionen überwachen.
- Im Fall eines Fehlers eine alternative Reihe von Aktionen ausführen.

Die Testen-Aktion erstellt **Ausführen-** und **Bei Fehler-**Platzhalter für Aktionen. Alle Aktionen, die bei Auslösen des Triggers ausgeführt werden sollen, müssen in den Platzhalter **Ausführen** aufgenommen werden. Wenn bei Ausführung der Aktionen aus dem **Ausführen**-Platzhalter kein Fehler auftritt, werden außer ihnen keine weiteren Aktionen ausgeführt. Wenn jedoch ein Fehler auftritt, wird die Ausführung der Aktionen im **Ausführen**-Platzhalter unterbrochen und die Aktionen im Platzhalter **Bei Fehler** werden ausgeführt.

ANMERKUNG

Sie müssen Synchrones Drucken aktivieren, um Fehler mit **Bei Fehler** zu erkennen.



Beispiel

Wenn eine Aktion im Platzhalter „Ausführen“ fehlschlägt, wird ihre Ausführung abgebrochen, und stattdessen werden die Aktionen im Platzhalter „Bei Fehler“ ausgeführt. Würde „Testen“ allein stehen, würde dies die Trigger-Ausführung beenden. In unserem Fall ist „Testen“ unter der Aktion „FOR Schleife“ eingebunden. Normalerweise würde ein Fehler im Platzhalter „Ausführen“ auch die Ausführung der Aktion „FOR Schleife“ beenden, auch wenn noch Schritte im Rahmen von „FOR Schleife“ ausstehen. In diesem Fall wird auch die Aktion „Daten in Datei speichern“ nicht ausgeführt. Standardmäßig unterbricht jeder Fehler die gesamte Trigger-Verarbeitung.

Sie können jedoch auch mit der Ausführung des nächsten Schritts in der Aktion „FOR Schleife“ fortfahren. Aktivieren Sie zu diesem Zweck die Option „Fehler ignorieren“ in der „Testen“-Aktion. Falls die Daten aus dem aktuellen Schritt in „FOR Schleife“ einen Fehler im „Ausführen“-Platzhalter verursachen, werden die Aktionen aus „Bei Fehler“ ausgeführt. Danach wird die Aktion „Daten in Datei speichern“ auf Ebene 2 ausgeführt, woraufhin die „FOR Schleife“-Aktion mit der Ausführung im nächsten Schritt fortfährt.

TIPP

Diese Aktion ermöglicht eine einfache Fehlererkennung sowie die Ausführung der Aktionen „Feedback“ oder „Berichterstattung“. Wenn zum Beispiel bei der Trigger-Verarbeitung ein Fehler auftritt, kann eine Warnmeldung gesendet werden. Weitere Informationen finden Sie im Abschnitt „Feedback zum Status von Druckaufträgen“ im NiceLabel Automation Benutzerhandbuch.

ANMERKUNG

Wichtig! Die **Testen**-Aktion liefert erwartungsgemäße Ergebnisse mit asynchronen Aktionen. Wenn Ihre Testen-Schleife die Aktion Etikett drucken enthält und diese fehlschlägt, führt die Aktion die Testen-Schleife dennoch vollständig aus und wechselt nicht, wie erwartet, zu den Aktionen **Bei Fehler**. Das Ergebnis des nicht erfolgten Wechsels zu den Aktionen **Bei Fehler** ist die Aktion Etikett drucken, die standardmäßig im synchronen Modus ausgeführt wird. Um dies zu vermeiden, stellen Sie sicher, dass überwachtes Drucken aktiviert ist. Öffnen Sie die Trigger-Einstellungen > **Andere** > **Feedback von der Print Engine** und aktivieren Sie **Überwachtes Drucken**.

Weitere Informationen über überwachtes Drucken finden Sie im Kapitel Synchroner Druckmodus.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  ANMERKUNG

  Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.15.6. XML-Umwandlung

PRODUKTEBENEN-INFO

Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

Diese Aktion wandelt ein XML-Dokument anhand der angegebenen Umwandlungsregeln in ein anderes Dokument um. Die Regeln müssen durch eine .XSLT-Definition in einer Datei oder in Form einer anderen variablen Quelle angegeben werden.

Mit dieser Aktion können Sie komplexe XML-Dokumente in XML-Dokumente mit übersichtlicherer Struktur konvertieren. XSLT steht für „XSL Transformations“. XSL steht für „Extensible Stylesheet Language“, eine Stylesheet-Sprache für XML-Dokumente.

Die Aktion „XML-Umwandlung“ speichert das konvertierte XML-Dokument in der ausgewählten Variablen. Die Originaldatei bleibt unberührt auf der Festplatte erhalten. Wenn Sie das konvertierte XML-Dokument speichern möchten, verwenden Sie die Aktion Daten in Datei speichern.

ANMERKUNG

Normalerweise verwenden Sie die Aktion, um von der Host-Anwendung bereitgestellte XML-Dokumente zu vereinfachen. Die Definition eines XML-Filters für das komplexe XML-Dokument würde eventuell zu lang dauern, und in einigen Fällen ist es sogar einfach zu komplex für die Verarbeitung. Als Alternative können Sie Regeln für die Konvertierung des XML-Dokuments in eine Struktur festlegen, die vom XML-Filter problemlos verarbeitet werden kann oder den Filter sogar völlig überflüssig macht. Sie können XML-Dokumente in ein nativ unterstütztes XML-Format wie Oracle XML umwandeln und dann einfach die Aktion Oracle XML-Befehlsdatei ausführen darauf anwenden.

TIPP

Ein Beispiel für diese Aktion wird mit dem Produkt installiert. Um es zu öffnen, gehen Sie auf **Hilfe** > **Beispieldateien** > **XML-Umwandlungen** und führen Sie die Konfiguration Transformations.misx aus. Genauere Angaben finden Sie in der **Readme**-Datei.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Datenquelle** definiert die umzuwandelnden XML-Daten.

- **Vom Trigger empfangene Daten verwenden:** legt fest, dass die vom Trigger empfangenen Daten verwendet werden. Dasselbe Ergebnis kann erzielt werden, indem Sie die interne Variable `DataFileName` aktivieren und die Inhalte der Datei verwenden, auf die sie verweist. Weitere Informationen finden Sie im Abschnitt „Zusammengesetzte Werte verwenden“ im NiceLabel Automation Benutzerhandbuch.
- **Dateiname:** definiert den Pfad und den Namen der umzuwandelnden XML-Datei. Der Inhalt der angegebenen Datei wird verwendet. Die Option Datenquelle aktiviert die dynamische Definition des Dateinamens. Wählen Sie eine Variable aus (oder erstellen Sie eine Variable), die den Pfad und/oder

Dateinamen enthält. Die Aktion öffnet die angegebene Datei und wendet die Umwandlung auf den (XML-formatierten) Dateiinhalt an.

- **Variable:** wählt die Variable aus, die den Druckdatenstrom enthält (oder erstellt diese Variable). Der Inhalt der ausgewählten Variablen wird verwendet und muss eine XML-Struktur aufweisen.

De Gruppe **Umwandlungsregeln Datenquelle (XSLT)** definiert die Umwandlungsregeln (.XSLT-Dokument), die auf das XML-Dokument angewandt werden.

- **Dateiname:** definiert den Pfad und den Namen der Datei mit den Umwandlungsregeln (.XSLT).
- **Benutzerdefiniert:** legt einen benutzerdefinierten Inhalt fest. Sie können festen Inhalt, eine Mischung aus festem und variablem Inhalt oder ausschließlich variablen Inhalt verwenden. Um variablen Inhalt einzufügen, klicken Sie auf die Schaltfläche mit dem Pfeil rechts neben dem Datenbereich und fügen Sie die Variable aus der Liste ein. Weitere Informationen finden Sie im Abschnitt „Zusammengesetzte Werte verwenden“ im NiceLabel Automation Benutzerhandbuch.

Die Gruppe **Ergebnis in Variable speichern** definiert die Variable, in der die umgewandelte Datei gespeichert werden soll.

- **Variable:** wählt eine Variable aus (oder erstellt eine Variable), die das Ergebnis des Umwandlungsprozesses beinhalten wird. Zum Beispiel: Wenn Sie Regeln verwenden, die eine komplexe XML-Struktur in eine einfachere XML-Struktur umwandeln, ist diese einfachere Struktur der Inhalt der ausgewählten Variablen.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

> **ANMERKUNG**
> Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.15.7. Gruppe

Diese Aktion konfiguriert mehrere Aktionen innerhalb desselben Containers. Alle unter der **Gruppe**-Aktion platzierten Aktionen gehören zur selben Gruppe und werden gemeinsam ausgeführt.

Diese Aktion bietet die folgenden Vorteile:

- **Bessere Organisation und Anzeige des Aktionsablaufs.** Sie können die Gruppenaktion erweitern oder minimieren, um die geschachtelten Aktionen nur bei Bedarf anzuzeigen. So halten Sie den Konfigurationsbereich aufgeräumt und übersichtlich.
- **Bedingungsabhängige Ausführung festlegen.** Sie können eine Bedingung in der Gruppe-Aktion nur einmal festlegen, nicht einzeln für jede Aktion. Wenn die Bedingung erfüllt ist, werden alle Aktionen innerhalb der Gruppe ausgeführt. Dies kann Ihnen eine Menge Konfigurationszeit sparen und die Anzahl von Konfigurationsfehlern mindern. Die Gruppenaktion ist optimal geeignet, um WENN...DANN-Ausführungsregeln für mehrere Aktionen festzulegen.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.15.8. Ereignis protokollieren

> **PRODUKTEBENEN-INFO**
> Automation Builder-Funktionen erfordern **NiceLabel LMS Enterprise** oder **NiceLabel LMS Pro**.

Diese Aktion protokolliert ein Ereignis zu Verlaufs- und Fehlerbehebungs-Zwecken in NiceLabel Control Center.

ANMERKUNG

Um die Aktion „Ereignis protokollieren“ zu aktivieren, stellen Sie sicher, dass die Druckauftrags-Protokollierung in NiceLabel Control Center aktiviert ist.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Ereignisdaten** bietet Informationen zum protokollierten Ereignis.

- **Information:** Allgemeine Beschreibung des Ereignisses, die in das NiceLabel Control Center Ereignisprotokoll aufgenommen wird. In diesem Bereich sind bis zu 255 Zeichen erlaubt.
- **Details:** Detaillierte Beschreibung des Ereignisses, das in NiceLabel Control Center protokolliert wird. In diesem Bereich sind bis zu 2000 Zeichen erlaubt.

TIPP

Die in den Feldern **Information** und **Details** eingegebenen Beschreibungen ermöglichen es Ihnen, die Ereignisse im **Gesamten Aktivitätsverlauf** von Control Center herauszufiltern. Öffnen Sie in Control Center **Verlauf** > **Alle Aktivitäten** > **Filter definieren**. Weitere Details finden Sie im Control Center Benutzerhandbuch.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

5.15.9. Etikettenvorschau

> **PRODUKTEBENEN-INFO**
> Automation Builder-Funktionen erfordern **LMS Enterprise**.

Diese Aktion führt den Druckprozess aus und stellt eine Vorschau des Etiketts bereit. Standardmäßig wird die Vorschau als JPEG auf der Festplatte gespeichert, aber Sie können auch andere Bildformate auswählen. Zudem können Sie die Größe des erstellten Vorschaubildes beeinflussen. Die Aktion erstellt eine Vorschau für ein einzelnes Etikett.

Nachdem Sie die Etikettenvorschau als Datei erstellt haben, können Sie diese an eine Drittanwendung senden, indem Sie eine der Übertragungsoptionen nutzen, z. B. Daten an HTTP senden, Daten an serielle Schnittstelle senden oder Daten an TCP/IP Port senden. Alternativ können Sie die Vorschaudatei auch als Antwortnachricht für bidirektionale Triggern nutzen, z. B. Webdienst-Trigger. Die Drittanwendung kann Benutzern die Bilddatei als Etikettenvorschau anzeigen.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Vorschau** definiert die Datei, für die die Vorschau erzeugt werden soll, sowie ihre Details.

- **Dateiname:** Gibt den Pfad und den Dateinamen an. Ist er fest codiert, wird jedes Mal dieselbe Datei verwendet. Wenn Sie nur den Dateinamen ohne Pfad verwenden, wird der Ordner mit der Konfigurationsdatei (.MISX) genutzt. Sie können eine relative Referenz zum Dateinamen verwenden, wobei der Ordner mit der .MISX-Datei als Stammverzeichnis fungiert. Die **Datenquelle**-Option aktiviert den variablen Dateinamen. Wählen Sie eine Variable aus (oder erstellen Sie eine Variable), die den Pfad und/oder Dateinamen bereitstellt, wenn ein Trigger ausgeführt wird. Normalerweise wird der Wert der Variablen durch einen Filter zugewiesen.
- **Grafiktyp:** legt den Grafiktyp fest, der zum Speichern der Etikettenvorschau verwendet wird.
- **Vorschau der Etikettenrückseite (zweiseitige Etiketten):** aktiviert die Vorschau der Rückseite des Etiketts. Dies ist nützlich, wenn Sie zweiseitige Etiketten verwenden und eine Vorschau der Rückseite des Etiketts erstellen möchten.

Beispiel

Zum Beispiel: Wenn Ihre Etikettenvorlage Abmessungen von 4" x 3" vorgibt und der Etikettendrucker eine Auflösung von 200 DPI hat, erhält das resultierende Vorschaubild Abmessungen von 800 x 600 Pixeln. Die Breite ist gleich 4 Zoll mal 200 DPI, also 800 Pixel. Die Höhe ist gleich 3 Zoll mal 200 DPI, also 600 Pixel.

Die Gruppe **Zusätzliche Einstellungen** ermöglicht es Ihnen, die Nutzung vorläufiger Werte zu aktivieren.

- **Vorläufige Werte verwenden:** ersetzt fehlende Datenquellenwerte durch vorläufige Werte und zeigt diese in der Etikettenvorschau an.

TIPP

Vorläufiger Wert legt beim Gestalten von Etiketten oder Masken einen benutzerdefinierten Platzhalter-Variablenwert in einem Objekt fest. In einem Etikettenobjekt wird der vorläufige Wert zum Druckzeitpunkt durch den echten Variablenwert ersetzt.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

  > **ANMERKUNG**
  > Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

5.15.10. Etikettenvariante erstellen

> **PRODUKTEBENEN-INFO**
> Automation Builder-Funktionen erfordern **LMS Enterprise**.

Diese Aktion ermöglicht Ihnen das Erstellen einer vorschaufertigen Variante eines vorhandenen Etiketts. Etikettenobjekte in solchen Varianten haben gesperrte Datenquellenwerte. Ihr Inhalt wird vom aktuellen Wert der jeweiligen Datenquelle bestimmt.

Der Sinn der Erstellung einer vorschaufertigen Variante eines Etiketts mit „gesperrten“ Datenquellen besteht darin, das Etikett für den Genehmigungsprozess vorzubereiten, bei dem Daten und Vorlage

gemeinsam genehmigt werden müssen. Anstatt ein Etikett ohne definierten Inhalt für seine Objekte anzuzeigen, genehmigt die zuständige Person eine Variante mit definierten Werten. So kann sie das finale Etikettenlayout mit den tatsächlichen Werten, die für den Druck verwendet werden, schnell anzeigen und genehmigen.

TIPP

Der Etikettengenehmigungsprozess ist für Etiketten möglich, die im Control Center Dokumentenspeicher abgelegt sind. Sie können verschiedene Arten von Genehmigungs-Workflows für die gespeicherten Etiketten und Etikettenvarianten anwenden. Die Auswahl für den Genehmigungs-Workflow hängt von den Anforderungen Ihrer Geschäftsumgebung ab. Im NiceLabel 10 Control Center Benutzerhandbuch finden Sie weitere Details.

Die **Über**-Gruppe gibt die ausgewählte Aktion an.

- **Name:** ermöglicht es Ihnen, einen benutzerdefinierten Namen anzugeben. So können Sie Aktionen in der Liste der Aktionen in der Lösung leichter erkennen. Standardmäßig werden Aktionsnamen vom Typ der jeweiligen Aktion abgeleitet.
- **Beschreibung:** benutzerdefinierte Informationen über die Aktion. Geben Sie eine Beschreibung ein, um den Zweck und die Rolle einer Aktion in einer Lösung zu erklären.
- **Aktionstyp:** Nur-Lesen-Informationen über den ausgewählten Aktionstyp.

Die Gruppe **Einstellungen** legt die zu konvertierende Etikettendatei und die Ausgabedatei (Etikettenvariante) fest.

- **Etikettenname:** der Name der Etikettendatei, die in eine prüfungsfertige Variante mit gesperrten Datenquellenwerten konvertiert werden soll. **Datenquelle** definiert den **Etikettennamen** anhand einer vorhandenen oder neu erstellten Variablen dynamisch.
- **Datenquellen zum Druckzeitpunkt:** mit dieser Option können Sie Datenquellen festlegen, deren Werte zum tatsächlichen Druckzeitpunkt bereitgestellt werden. Wenn in diesem Feld eine Datenquelle aufgeführt ist, wird ihr Wert nicht gesperrt und kann zum Druckzeitpunkt bereitgestellt werden. Typische Beispiele sind Datenquellen für Produktionswerte wie Losnummer, Verfallsdatum usw.

  TIPP

  Geben Sie nur Datenquellennamen ohne eckige Klammern ein, entweder durch Kommas getrennt oder anhand der Eingabetaste in Spalten unterteilt.

- **Name der Ausgabedatei:** der Name der Etikettenvariante, die für die Prüfung und Genehmigung erstellt wird. **Datenquelle** definiert den **Etikettennamen** anhand einer vorhandenen oder neu erstellten Variablen dynamisch.

Es gibt verschiedene Regeln, die für die prüfungsfertige Etikettenvariante gelten:

1. Datenquellenwerte sind standardmäßig gesperrt. Um Datenquellen von der Sperre auszunehmen, listen Sie sie im Feld **Datenquellen zum Druckzeitpunkt** auf, damit sie auf dem prüfungsfertigen Etikett aktiv bleiben. Sie können ihre Werte zum Druckzeitpunkt festlegen.
2. Zählervariablen, Funktionen, Datenbankfelder und globale Variablen werden in nicht-abgefragte Variablen konvertiert.
3. Grafiken werden eingebettet.
4. Die im Dokumentenspeicher von NiceLabel Control Center abgelegte Ziel-Etikettenvariante wird automatisch eingecheckt. Der ursprüngliche **Etikettenname** und die **Datenquellen zum Druckzeitpunkt** werden als Check-in-Kommentar genutzt.
5. Etikettenvarianten können im gesperrten Zustand im NiceLabel 10 Designer geöffnet werden.
6. Anhand dieser Aktion erzeugte Etikettendateien können nicht importiert werden.
7. Wenn Etikettenvarianten im Druckerspeicher abgelegt werden, können anhand des Abrufbefehls nur Werte für Druckzeitpunkt-Datenquellen bereitgestellt werden.
8. Wenn Sie NiceLabel Control Center verwenden, ermöglicht die Etikettenvorschau im Dokumentenspeicher die Bearbeitung von Druckzeitpunkt-Datenquellen.
9. Variablen für die aktuelle Uhrzeit und das aktuelle Datum können auf der prüfungsfertigen Etikettenvariante nicht als Datenquellen zum Druckzeitpunkt festgelegt werden.

Aktionsausführung und Fehlerhandhabung

Jede Aktion kann als bedingungsabhängige Aktion festgelegt werden. Bedingungsabhängige Aktionen werden nur ausgeführt, wenn die festgelegten Bedingungen erfüllt sind. Um diese Bedingungen zu definieren, klicken Sie auf **Optionen für Ausführung und Fehlerhandhabung anzeigen**.

Die **Ausführungsoptionen** sind:

- **Aktiviert:** gibt an, ob die Aktion aktiviert oder deaktiviert ist. Nur aktivierte Aktionen werden ausgeführt. Diese Funktion kann beim Testen einer Druckmaske verwendet werden.
- **Bedingung:** definiert einen einzeiligen Programmierausdruck, welcher einen booleschen Wert bereitstellen muss (`wahr` oder `falsch`). Ist das Ergebnis des Ausdrucks `wahr`, wird die Aktion ausgeführt. Bedingungen sorgen dafür, dass Aktionen nicht jedes Mal ausgeführt werden müssen.

Die **Fehlerhandhabung**-Optionen sind:

- **Fehler ignorieren:** gibt an, ob ein Fehler ignoriert werden soll oder nicht. Wenn die Option **Fehler ignorieren** aktiviert ist, wird die Ausführung von Aktionen selbst dann fortgesetzt, wenn die aktuelle Aktion fehlschlägt.

ANMERKUNG

Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht zu einer Unterbrechung der Ausführung von Aktionen.

Beispiel

Am Ende des Druckvorgangs möchten Sie eventuell mittels der Aktion **HTTP-Anfrage** einen Statusbericht an eine externe Anwendung senden. Wenn die Druckaktion fehlschlägt, wird die Aktionsverarbeitung angehalten. Um die Berichterstellung trotz fehlgeschlagener Druckaktion auszuführen, muss für die Aktion **Etikett drucken** die Option **Fehler ignorieren** aktiviert sein.

- **Fehler in Variable speichern**: ermöglicht Ihnen, eine Variable auszuwählen oder zu erstellen, in der der Fehler gespeichert werden soll. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen **`ActionLastErrorId`** und **`ActionLastErrorDesc`** gespeichert.

6. Trigger ausführen und verwalten

6.1. Konfiguration anwenden

Wenn Sie die Trigger in Automation Builder konfiguriert und getestet haben, stellen Sie die Konfiguration anhand des NiceLabel Automation Dienstes bereit und starten die Trigger. Daraufhin werden die Trigger aktiv und beginnen mit der Überwachung der festgelegten Ereignisse.

Verwenden Sie eine der folgenden Methoden, um die Konfiguration anzuwenden.

Implementierung aus Automation Builder

1. Starten Sie Automation Builder.
2. Laden Sie die Konfiguration.
3. Öffnen Sie die Registerkarte **Konfigurationselemente**.
4. Klicken Sie auf die Schaltfläche **Konfiguration anwenden** in der Gruppe „Anwenden“.
   Die Konfiguration wird im Automation Manager auf demselben Rechner ausgeführt.
5. Starten Sie die Trigger, die Sie aktivieren möchten.

Wenn diese Konfiguration bereits geladen wurde, erzwingt die Implementierung ein erneutes Laden, hält aber den Status des Triggers aktiv.

Implementierung aus Automation Manager

1. Starten Sie Automation Manager.
2. Öffnen Sie die Registerkarte **Trigger**.
3. Klicken Sie auf die **+Hinzufügen**-Schaltfläche und navigieren Sie auf der Festplatte zur Konfiguration.
4. Starten Sie die Trigger, die Sie aktivieren möchten.

Implementierung anhand der Befehlszeile

Um die Konfiguration **`C:\Project\Configuration.MISX`** anzuwenden und den darin enthaltenen Trigger namens **`CSVTrigger`** anhand der Befehlszeile auszuführen, geben Sie Folgendes ein:

```
NiceLabelAutomationManager.exe ADD c:\Project\Configuration.MISX
NiceLabelAutomationManager.exe START c:\Project\Configuration.MISX CSVTrigger
```

Weitere Informationen finden Sie im Abschnitt Dienst mit Befehlszeilenparametern steuern.

6.2. Ereignisprotokoll-Optionen

PRODUKTEBENEN-INFO

Einige der in diesem Abschnitt beschriebenen Funktionen erfordern den Kauf von **NiceLabel LMS** Produkten.

NiceLabel Automation protokolliert Ereignisse an verschiedenen Orten, die vom Implementierungs-Szenario abhängen. Die ersten beiden Protokollfunktionen sind in jeder NiceLabel Automation Produktebene verfügbar.

- **Protokollierung in der Log-Datenbank:** Die Protokollierung in der internen Log-Datenbank ist immer aktiviert. In der internen Log-Datenbank werden Datensätze aller Ereignisse mit allen Details gespeichert. Bei der Anzeige der protokollierten Informationen können Sie Filter verwenden, um Ereignisse anzuzeigen, die den Regeln entsprechen. Weitere Informationen finden Sie im Abschnitt Ereignisprotokoll verwendenn im Benutzerhandbuch.
  Die Daten werden in einer SQLite-Datenbank gespeichert. Dieser Speicher ist nur temporär: Die Ereignisse werden auf wöchentlicher Basis aus der Datenbank entfernt. Das Aufräumintervall kann in den Optionen konfiguriert werden. Datensätze mit alten Ereignissen werden aus der Datenbank gelöscht, aber die Datenbank wird nicht kompaktiert, sodass der Festplattenspeicher eventuell belegt bleibt. Verwenden Sie zum Kompaktieren der Datenbank eine SQLite-Verwaltungssoftware eines Drittanbieters.
- **Protokollierung im Windows-Anwendungsereignisprotokoll:** Wichtige Ereignisse werden im Windows-Anwendungsereignisprotokoll gespeichert, falls NiceLabel Automation nicht startet. So wird gewährleistet, dass es eine zweite Ressource für protokollierte Ereignisse gibt.
- **Protokollierung im Control Center:** Die Protokollierung im Control Center ist in **LMS Enterprise** und **LMS Pro** Produkten verfügbar. Control Center ist eine webbasierte Management-Konsole, die alle Ereignisse auf einem oder mehreren NiceLabel Automation-Servern aufzeichnet. Die Daten werden in der Microsoft SQL Server-Datenbank gespeichert. Sie können die erhobenen Daten durchsuchen, und die Anwendung unterstützt außerdem automatisierte Meldungen zu bestimmten Ereignissen, Druckerverwaltung, Dokumentenspeicher, Revisionskontrolle (Versionierung), Workflows und erneuten Etikettendruck.

> **ANMERKUNG**
> Weitere Informationen finden Sie im Control Center Anwenderhandbuch.

6.3. Trigger verwalten

Automation Manager ist der Verwaltungsteil der NiceLabel Automation-Software. Wenn Sie Automation Builder zur Konfiguration der Trigger verwenden, nutzen Sie Automation Manager, um sie in der Produktionsumgebung bereitzustellen und auszuführen. Die Anwendung ermöglicht es Ihnen, Trigger aus verschiedenen Konfigurationen zu laden, ihren Status in Echtzeit zu verfolgen, sie zu starten/anzuhalten und Ausführungsdetails in der Log-Datei anzuzeigen.

Sie können die Ansicht der geladenen Konfigurationen und Trigger anpassen. Die letzte Ansicht wird gespeichert und angewandt, wenn Sie Automation Manager zum nächsten Mal verwenden. Wenn Sie die Anzeige **Nach Status** aktivieren, werden Trigger aus allen offenen Konfiguration, die denselben Status aufweisen, zusammen angezeigt. Wenn Sie die Anzeige nach **Konfigurationen** aktivieren, werden Trigger aus der ausgewählten Konfiguration angezeigt, unabhängig davon, welchen Status sie aufweisen. Der Trigger-Status wird zwecks einfacher Erkennung im Trigger-Symbol durch die jeweilige Farbe angezeigt.

Die angezeigten Trigger-Details ändern sich in Echtzeit, sobald die Trigger-Ereignisse erkannt werden. Sie sehen verschiedene Informations-Elemente, z. B. Name des Triggers, Typ des Triggers, die Anzahl der bereits verarbeiteten Ereignisse, die Anzahl der erkannten Fehler und die vergangene Zeit seit dem letzten Ereignis. Wenn Sie mit dem Mauszeiger über die Anzahl der bereits verarbeiteten Trigger fahren, sehen Sie, wie viele Trigger auf die Verarbeitung warten.

> **ANMERKUNG**
> Die geladene Konfiguration wird im Arbeitsspeicher abgelegt. Wenn Sie eine Änderung an der Konfiguration im Automation Builder vornehmen, wendet der Automation Manager sie nicht automatisch an. Um die Änderung anzuwenden, laden Sie die Konfiguration neu.

Konfiguration laden

Um eine Konfiguration neu zu laden, klicken Sie auf die **+Hinzufügen**-Schaltfläche und suchen Sie nach der Konfigurationsdatei (.MISX). Trigger aus der Konfiguration werden im inaktiven Status geladen. Um die Trigger zu aktivieren, müssen Sie sie starten. Weitere Informationen finden Sie im Abschnitt Konfiguration anwenden.

Die Liste geladener Konfigurationen und Status für jeden Trigger werden gespeichert. Wird der Server aus einem beliebigen Grund neu gestartet, stellt der NiceLabel Automation-Dienst den Status des Triggers vor dem Neustart wieder her.

Konfiguration neu laden und entfernen

Nach Aktualisieren und Speichern der Konfiguration in Automation Builder werden die Änderungen in Automation Manager nicht automatisch angewandt. Um die Konfiguration neu zu laden, klicken Sie mit der rechten Maustaste auf den Konfigurationsnamen und dann auf **Konfiguration neu laden**. Dadurch werden alle Trigger neu geladen. Wenn Sie Datei-Caching aktiviert haben, wird durch das erneute Laden eine Synchronisierung aller Dateien erzwungen, die von den Triggern verwendet werden.

Trigger starten/anhalten

Wenn Sie Trigger aus einer Konfiguration laden, befinden sie sich standardmäßig im angehaltenen Status. Um den Trigger zu starten, klicken Sie auf die Schaltfläche **Start** im Trigger-Bereich. Um den Trigger anzuhalten, klicken Sie auf **Stopp**. Sie können weitere Trigger aus derselben Konfiguration auswählen und alle davon gleichzeitig starten/anhalten.

Außerdem können Sie das Starten/Anhalten einer Konfiguration über die Befehlszeile steuern. Weitere Informationen finden Sie im Abschnitt Dienst mit Befehlszeilenparametern steuern.

Trigger-Konflikte handhaben

Trigger können aufgrund der im Folgenden aufgeführten Situationen einen Fehlerstatus aufweisen. Sie können Trigger mit Fehlern erst starten, wenn Sie das Problem behoben haben.

- **Trigger nicht korrekt oder vollständig konfiguriert:** In diesem Fall ist der Trigger nicht konfiguriert, obligatorische Eigenschaften sind nicht definiert oder für den jeweiligen Drucker festgelegte Aktionen sind nicht konfiguriert. Sie können solche Trigger nicht starten.
- **Trigger-Konfiguration überschneidet sich mit anderem Trigger:** Zwei Trigger können nicht ein und dasselbe Ereignis überwachen.

Beispiel

Zwei Datei-Trigger können nicht ein und dieselbe Datei überwachen. Zwei HTTP-Trigger können keine Daten an derselben Schnittstelle empfangen. Falls sich die Trigger-Konfiguration mit einem anderen Trigger überschneidet, kann der zweite Trigger nicht ausgeführt werden, da das Ereignis bereits vom ersten Trigger erfasst wurde. Weitere Informationen finden Sie im Protokoll-Bereich für den entsprechenden Trigger.

Fehlerstatus zurücksetzen

Wenn die Ausführung des Triggers einen Fehler hervorruft, nimmt das Trigger-Symbol eine rote Farbe an, der Trigger erhält einen Fehlerstatus und die Ereignisdetails werden in der Protokolldatenbank erfasst. Selbst wenn alle folgenden Ereignisse erfolgreich abgeschlossen werden, verbleibt der Trigger im Fehlerstatus, bis Sie bestätigen, dass Sie den Fehler zur Kenntnis genommen haben und den Status entfernen möchten. Um den Fehler zu bestätigen, klicken Sie auf das Symbol neben dem Fehlerzähler unter den Trigger-Details.

Meldungsbereich verwenden

Der Meldungsbereich ist der Bereich über der Liste von Triggern auf der Registerkarte „Trigger“, wo wichtige Meldungen angezeigt werden. Im Meldungsbereich werden **Statusmeldungen** zur Anwendung wie z. B. „Testmodus“ oder „Testmodus abgelaufen“ oder **Warnmeldungen** wie „Verfolgung wurde aktiviert“ angezeigt.

Protokollierte Daten anzeigen

Alle Trigger-Aktivitäten werden in der Datenbank protokolliert, darunter Start/Stopp-Ereignisse, erfolgreiche Ausführungen von Aktionen und bei der Verarbeitung aufgetretene Fehler. Klicken Sie auf die Schaltfläche „Log“, um protokollierte Ereignisse für den ausgewählten Trigger anzuzeigen. Weitere Informationen finden Sie im Abschnitt Ereignisprotokoll verwendenn im Benutzerhandbuch.

6.4. Ereignisprotokoll verwenden

Alle Aktivitäten in NiceLabel Automation werden in einer Datenbank protokolliert, um Verlaufsdaten bereitzustellen und die Fehlerbehebung zu erleichtern. Wenn Sie auf die **Log**-Schaltfläche im Trigger-Tab klicken, werden Ereignisse für den jeweiligen Trigger angezeigt. Im Protokollbereich werden Informationen für alle Ereignisse dargestellt, die mit dem definierten Filter in Verbindung stehen.

Protokolldaten sind für die Fehlerbehebung hilfreich. Kann ein Trigger oder eine Aktion nicht ausgeführt werden, zeichnet die Anwendung eine Fehlerbeschreibung in der Protokolldatei auf, mithilfe derer Sie das Problem erkennen und lösen können.

ANMERKUNG

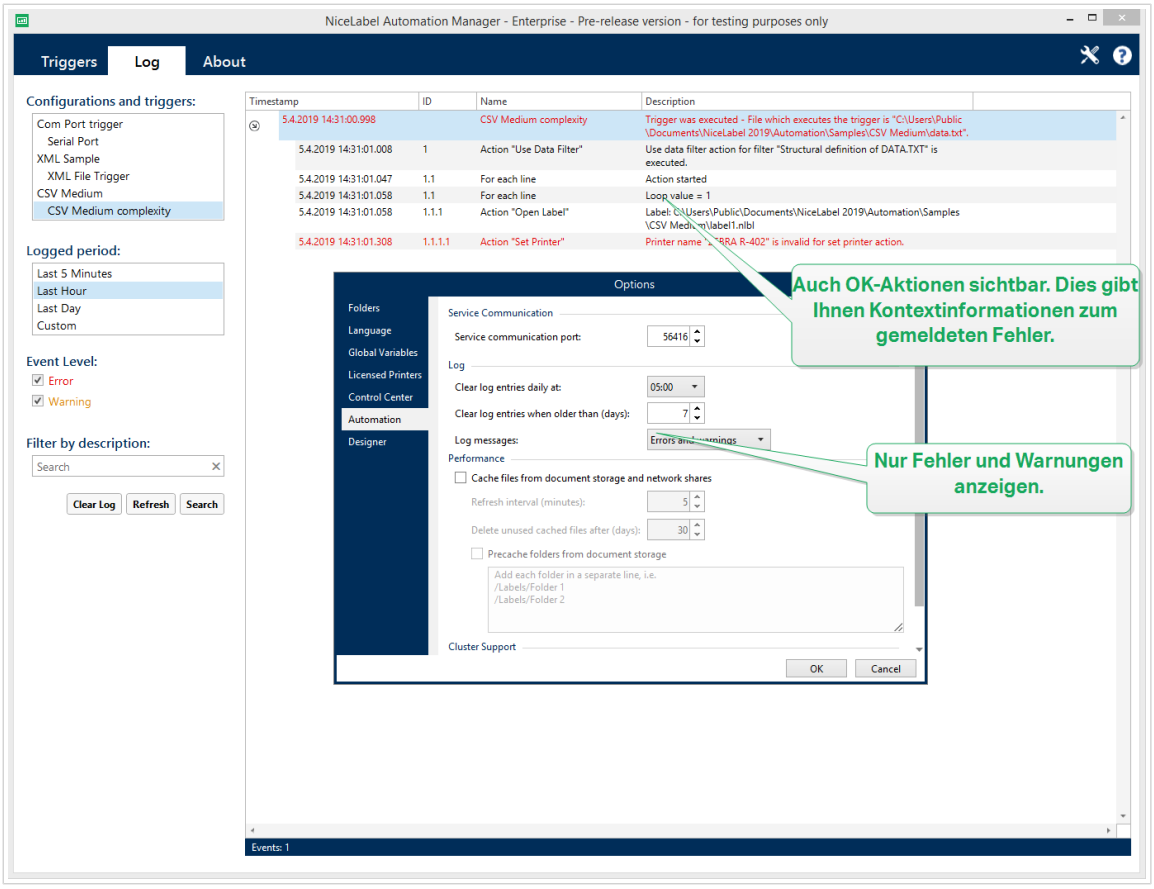
Die Aufbewahrungszeit für Daten beträgt standardmäßig 7 Tage und kann in den Optionen eingestellt werden. Um die Größe der Datenbank in ausgelasteten Systemen zu minimieren, können Sie die Aufbewahrungszeit reduzieren.

Ereignisse filtern

Die konfigurierbaren Filter:

- **Konfigurationen und Trigger:** Legt fest, welche Ereignisse angezeigt werden sollen – Ereignisse vom ausgewählten Trigger oder Ereignisse von allen Triggern in der ausgewählten Konfiguration.
- **Logging-Zeitraum:** Legt den Zeitrahmen fest, in dem Ereignisse eingetreten sind. Der Standard-Zeitraum ist **Letzte 5 Minuten**.
- **Ereignisebene:** Gibt den Typ (Bedeutung) der Ereignisse an, die Sie anzeigen möchten:
  - **Fehler** ist der Ereignistyp, der die Ausführung unterbricht.
  - **Warnung** ist der Ereignistyp, bei dem Fehler passieren, aber per Konfiguration ignoriert werden sollen.
  - **Information** ist der Ereignistyp, der alle Informationen protokolliert, die nicht mit Fehlern in Zusammenhang stehen.

> **ANMERKUNG**
> Im Fall von Fehlern und Warnungen zeigt Automation auch die gesamte Abfolge von erfolgreich ausgeführten Aktionen in einem Trigger an.



Die Protokollebene kann in den **Optionen** konfiguriert werden.

- **Filtern nach Text:** Sie können alle Ereignisse anzeigen, die die angegebene Zeichenfolge enthalten. Verwenden Sie diese Option, um Fehler in Triggern mit großem Textumfang zu beheben. Der Filter wird auf das Beschreibungsfeld des Triggers angewandt.

Protokolldatenbank leeren

Sie können das Protokoll aus Automation Builder löschen. Um die Protokolldatenbank zu löschen, klicken Sie auf die Schaltfläche **Log löschen**.

WARNUNG

Seien Sie beim Löschen von Protokollen vorsichtig, da dieser Befehl nicht rückgängig gemacht werden kann. Durch Ausführen von „Log löschen“ werden **ALLE** protokollierten Ereignisse aus der Datenbank gelöscht, und die Option gilt für alle Trigger, nicht nur für den aktuellen.

Automatisierte Bereinigungen des Protokolldatenbank

Automation ermöglicht es Ihnen, regelmäßiges automatisiertes Löschen für Protokolleinträge zu erfolgreich ausgeführten Triggern einzurichten. Auf diese Weise stellen Sie sicher, dass die wachsende Protokolldatenbank nicht die Systemleistung schmälert.

So planen Sie automatisierte Bereinigungen der Protokolldatenbank:

1. Öffnen Sie die Datei **`product.config`** in einem Text-Editor.
   Die Datei befindet sich hier:

   ```
   %PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
   ```

2. Erstellen Sie eine Sicherungskopie der Datei **`product.config`**.

3. Automation verwendet zwei Parameter, um die Triggermeldungen zu löschen. Fügen Sie diese Parameter zu Ihrer Datei **`product.config`** hinzu.

   - **`/IntegrationService/LogSuccessfulTriggerPurgeInterval>`**. Dieser Parameter legt die Länge des Zeitintervalls zwischen zwei aufeinander folgenden Bereinigungen fest. Geben Sie die Intervalllänge in Minuten ein.
   - **`/IntegrationService/LogSuccessfulTriggerPurgeRemovalAge>`**. Dieser Parameter prüft das Alter der Nachrichten zu den erfolgreich ausgeführten Aktionen.

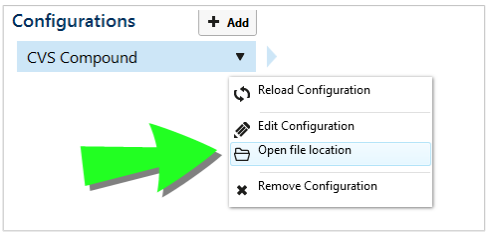
```
<configuration>
    <IntegrationService>
        <LogSuccessfulTriggerPurgeInterval>1</
LogSuccessfulTriggerPurgeInterval>

<LogSuccessfulTriggerPurgeRemovalAge>1<LogSuccessfulTriggerPurgeRemovalAg
e>
    </IntegrationService>
</configuration>
```

6.5. Wenn Ihre Konfiguration nicht geladen werden kann...

Nach Implementierung Ihrer Automation-Konfiguration wird diese als Windows-Prozess im Hintergrund ausgeführt. Der Automation Manager, mit dem Sie Ihre Konfiguration verwalten und überwachen, ist nur eine Oberfläche, die die eigentlichen Automation-Dienste darstellt. In bestimmten Fällen kann die Konfiguration, die Sie erstellt, getestet und implementiert haben, nicht geladen werden. Dafür gibt es mehrere mögliche Gründe. Folgen Sie den Lösungsvorschlägen, um Ihre Automation funktionsfähig zu machen:

1. Die Konfigurationsdatei wurde entfernt, umbenannt oder an einen anderen Speicherort verschoben. Sie können im Automation Manager prüfen, auf welche Datei Ihre implementierte Konfiguration verweist:

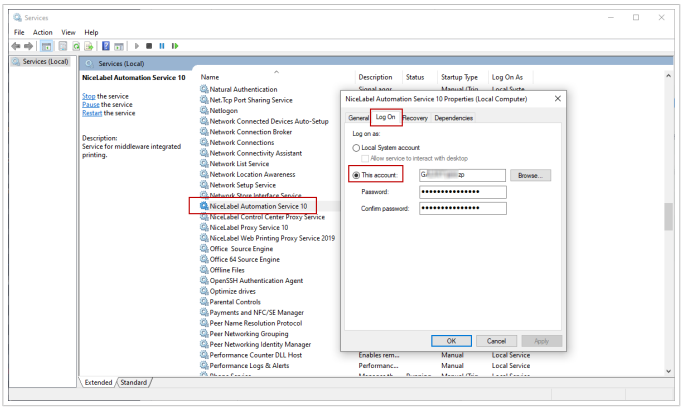


   Stellen Sie sicher, dass die .misx-Konfigurationsdatei am angegebenen Speicherort vorhanden ist und denselben Namen hat wie im Automation Manager. Falls Sie Ihre Konfigurationsdatei verschoben oder umbenannt haben, öffnen Sie sie im Automation Builder und implementieren Sie die Konfiguration erneut.

2. Die Konfigurationsdatei befindet sich an einem Speicherort im Netzwerk, auf den aufgrund eines Netzwerkverbindungsproblems nicht zugegriffen werden kann. Prüfen Sie die Netzwerkverbindung des Computers/Servers, auf dem die Konfiguration gespeichert ist.

3. **Der Automation-Dienst hat keine Berechtigung zum Zugriff auf die Konfigurationsdatei.** Prüfen Sie die Berechtigungen des Benutzerkontos, das der Automation-Dienst verwendet. Dieser Fehler deutet darauf hin, dass Probleme mit dem im Hintergrund ausgeführten Automation-Dienst bestehen.
   Mögliche Lösungen sind:

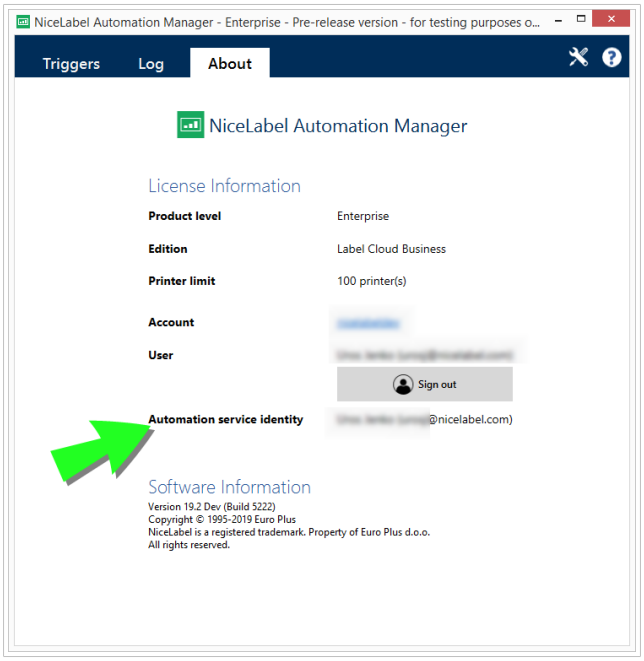
   - Falls Ihre Konfigurationsdatei auf einer lokalen Festplatte oder einer Netzwerkfreigabe gespeichert ist, sollten Sie sicherstellen, dass Sie Ihre Konfiguration mit den Zugangsdaten für Ihren lokalen Benutzer oder Ihre Domain ausführen. Eine Ausführung Ihrer Konfiguration unter dem lokalen Systemkonto kann dazu führen, dass der Zugriff auf freigegebene Netzwerkordner und -drucker nur eingeschränkt möglich ist. Öffnen Sie die Dienste und zeigen Sie die Eigenschaften von NiceLabelAutomation Service 10 an.



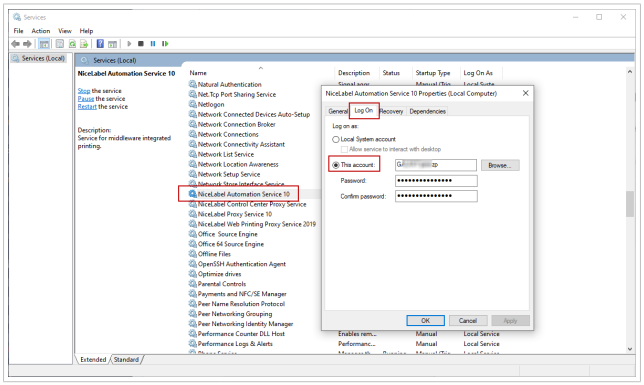
   - Wenn Ihre Konfigurationsdatei im Dokumentenspeicher Ihres Control Center gespeichert ist, sind dies die möglichen Szenarien:

     - Ihr Control Center nutzt die Anwendungs-Authentifizierung. Der Grund, dass Ihre Konfiguration nicht geladen werden kann, ist eine fehlerhafte **Automationsdienst-Identität**. Die Automationsdienst-Identität muss der in Ihrem Control Center definierten Benutzeridentität entsprechen.

Sie finden die Automationsdienst-Identität für Ihre Konfiguration unter **Automation Manager** > **Registerkarte Über**.



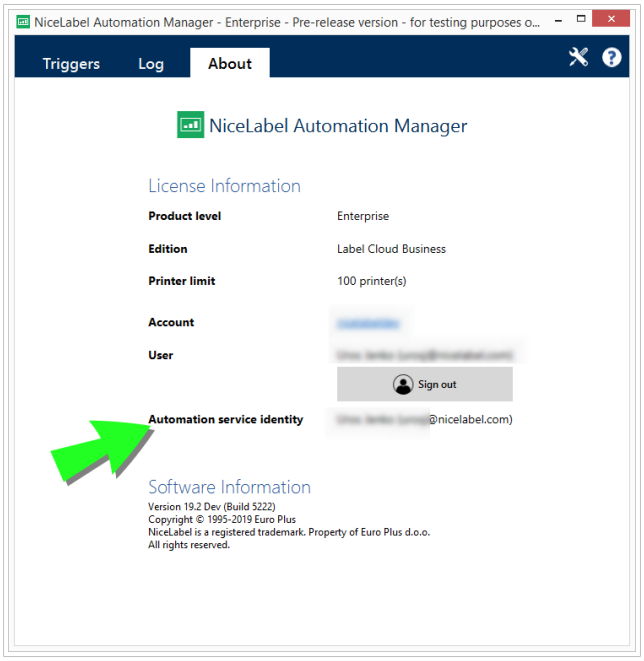
- Ihr Control Center nutzt die Windows-Authentifizierung. Ihre Konfiguration kann nicht geladen werden, weil Sie versuchen, sie als Benutzer mit ungenügenden Berechtigungen in Ihrem Control Center auszuführen. Prüfen Sie, mit welchem Benutzer Sie den Automation-Dienst ausführen. Öffnen Sie die **Dienste** und zeigen Sie die Eigenschaften von **NiceLabelAutomation Service 2019** an.



> **ANMERKUNG**
>
> In beiden Fällen finden Sie im Benutzerhandbuch für NiceLabelControl Center weitere Informationen zu den verfügbaren Authentifizierungsmethoden und Benutzerberechtigungen.

- Ihr Control Center wird in der NiceLabel Cloud ausgeführt. Der Grund, dass Ihre Konfiguration nicht geladen werden kann, ist eine fehlerhafte Automationsdienst-Identität. Die Automationsdienst-Identität muss der Benutzeranmeldung in der NiceLabel Cloud entsprechen.

Sie finden die **Automationsdienst-Identität** für Ihre Konfiguration unter **Automation Manager** > **Registerkarte Über**.



7. Performance- und Feedback-Optionen

7.1. Parallele Verarbeitung

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

NiceLabel Automation unterstützt parallele Verarbeitung sowohl bei der eingehenden als auch bei der ausgehenden Verarbeitung. Dies sorgt für maximale Effizienz auf allen System mit installierter Software. NiceLabel Automation führt mehrere Aufgaben gleichzeitig aus, behält aber dennoch die Reihenfolge bei, in der die Trigger hinzugefügt wurden. Der Durchsatz bei der Verarbeitung von Etikettenaufträgen hängt in hohem Maße von der verwendeten Hardware ab.

Eingehende parallele Verarbeitung

Sie können mehrere Trigger auf demselben Rechner ausführen. Alle von ihnen reagieren gleichzeitig auf Änderungen in den überwachten Ereignissen. Jeder Trigger speichert die Daten von nicht verarbeiteten Ereignissen in der Warteschlangenliste. Diese Liste puffert eingehende Daten für den Fall, dass momentan keiner der Druckprozesse verfügbar ist. Sobald einer der Druckprozesse verfügbar wird, wird der erste Auftrag anhand des FIFO-Prinzips (First In, First Out) aus der Warteschlange genommen. Dies gewährleistet die richtige Reihenfolge bei der eingehenden Datenverarbeitung. Es gewährleistet jedoch nicht, dass das FIFO-Prinzip auch beim Drucken angewandt wird. Siehe nächsten Abschnitt.

ANMERKUNG

Parallele Verarbeitung bedeutet mehr als nur das gleichzeitige Ausführen mehrerer Trigger. Jeder Trigger kann auch gleichzeitige Verbindungen eingehen. TCP/IP-, HTTP- und Webdienst-Trigger akzeptieren allesamt gleichzeitige Verbindungen von mehreren Clients. Außerdem kann der Datei-Trigger für die Überwachung einer Reihe von Dateien in einem Ordner konfiguriert werden. Dabei ist eine Konfiguration nach Dateimaske möglich.

Ausgehende parallele Verarbeitung

Normalerweise ist das Ergebnis des Triggers ein Etikettendruckvorgang. Für diesen Vorgang nutzen Sie vom Trigger empfangene Daten, um sie auf Etiketten zu drucken.Der NiceLabel Automation-Dienst führt die Druckvorgänge (auch „Druck-Engines“ genannt) im Hintergrund parallel aus. Moderne Prozessoren haben zwei oder mehr unabhängige Hauptprozessoren (CPUs), Kerne genannt. Mehrere Kerne können mehrere Anweisungen gleichzeitig ausführen, wodurch die gesamte Verarbeitungsgeschwindigkeit erhöht wird. Im Fall von NiceLabel Automation sorgen mehrere Kerne für eine schnellere Verarbeitung von Druckaufträgen und damit letztendlich für eine höhere Etikettendruckleistung.

Standardmäßig führt jede Instanz von NiceLabel Automation jeden Druckprozess als separaten Thread auf jedem verfügbaren Kern aus. Je leistungsstärker Ihre CPU ist, desto höher ist der verfügbare Durchsatz. So wird die verfügbare CPU-Leistung bestmöglich genutzt. Die Software wird mit angemessenen Standardwerten ausgeführt, die festlegen, dass jeder verfügbare Kern einen einzelnen Thread für die Druckverarbeitung übernimmt. Unter normalen Umständen ist es nicht nötig, Änderungen an den Standardwerten vorzunehmen. Wenn Ihre Konfiguration eine Änderung erfordert, finden Sie Informationen in Abschnitt Standardeinstellungen für Multi-Thread-Druck ändern.

Wenn mehrere Druckprozesse verfügbar sind, können die Daten aus dem ersten Ereignis über einen Druckprozess gedruckt werden, während die Daten aus dem zweiten Ereignis gleichzeitig über einen anderen Druckprozess gedruckt werden, sofern ein solcher zu diesem Zeitpunkt zur Verfügung steht. Falls das zweite Ereignis nur eine geringe Datenmenge bereitstellt, stellt der zweite Druckprozess die Daten für den Drucker eventuell schneller bereit als der erste Druckprozess, wodurch die Reihenfolge geändert wird. In einem solchen Fall ist es möglich, dass die Daten aus dem zweiten Ereignis vor den Daten aus dem ersten Ereignis gedruckt werden. Um sicherzustellen, dass das FIFO-Prinzip auch beim Drucken zum Einsatz kommt, siehe Abschnitt Synchroner Druckmodus.

7.2. Dateien zwischenspeichern

Um die Zeit bis zum ersten gedruckten Etikett und die Gesamtperformance zu optimieren, unterstützt NiceLabel Automation das Zwischenspeichern von Dateien. Wenn Sie Etiketten, Bilder und Datenbankdaten aus Netzwerkfreigaben laden, kann es zu Verzögerungen beim Druck Ihrer Etiketten kommen.NiceLabel Automation muss alle angeforderten Dateien abrufen, bevor der Druckprozess beginnen kann.

Es gibt zwei Ebenen der Zwischenspeicherung, die sich gegenseitig ergänzen.

- **Arbeitsspeichercache:** Im Arbeitsspeichercache werden die bereits verwendeten Dateien gespeichert. Etiketten, die mindestens einmal verwendet wurden, werden ins Arbeitsspeichercache geladen. Wenn ein Trigger einen Ausdruck solcher Etiketten anfordert, stehen sie sofort für den Druckprozess zur Verfügung. Das Arbeitsspeichercache ist standardmäßig aktiviert. Sein Inhalt wird gelöscht, nachdem Sie eine Konfiguration entfernen oder neu laden. Etikettendateien werden für jede „Etikett öffnen“-Aktion auf Änderungen geprüft. Wenn ein neueres Etikett zur Verfügung steht, wird es automatisch geladen und ersetzt die alte Version im Zwischenspeicher.

ANMERKUNG

Nachdem ein Etikett 8 Stunden lang nicht genutzt wurde, wird es aus dem Arbeitsspeichercache entfernt.

- **Permanenten Zwischenspeicher:** Der permanente Zwischenspeicher speichert Daten auf der Festplatte – seine Aufgabe besteht darin, mittelfristigen Speicher für Dateien zu bieten. Das Zwischenspeichern wird pro Dateiobjekt verwaltet. Nachdem eine Datei aus der Netzwerkfreigabe angefordert wurde, prüft der Dienst zuerst, ob die Datei bereits im Zwischenspeicher vorhanden ist; falls ja, wird diese Version verwendet. Falls nein, wird sie aus der Netzwerkfreigabe abgerufen und zur zukünftigen Nutzung zwischengespeichert. Der Zwischenspeicher-Dienst aktualisiert den Inhalt des Zwischenspeichers permanent mit neuen Versionen der Dateien. Sie können die Zeitintervalle für die Versionsprüfung im Optionen-Menü einstellen.

Zeit bis zur Entfernung von Etiketten aus dem Zwischenspeicher verlängern

Nachdem ein Etikett zum ersten Mal genutzt wurde, wird es ins Arbeitsspeichercache geladen. Wenn es das nächste Mal benötigt wird, steht es für sofortiges Drucken zur Verfügung. Alle Etiketten, die 8 Stunden lang oder länger nicht genutzt wurden, werden automatisch aus dem Zwischenspeicher entfernt.

So können Sie die Zeit bis zur Entfernung der Etiketten aus dem Zwischenspeicher verlängern:

1. Navigieren Sie zum NiceLabel Automation-Systemordner.
   `%PROGRAMDATA%\NiceLabel\NiceLabel 10`
2. Erstellen Sie eine Kopie der Datei **`product.config`**.
3. Öffnen Sie **`product.config`** in einem Text-Editor. Die Datei hat eine XML-Struktur.
4. Fügen Sie das Element **`Common/FileUpdater/PurgeAge`** hinzu.
5. Dieser Parameter definiert die Anzahl von Sekunden, die den Zeitrahmen für die Aufbewahrung von Etiketten im Arbeitsspeichercache bildet.NiceLabel Automation speichert die verstrichene Zeit nach dem letzten Drucken des Etiketts. Sobald der festgelegte Schwellenwert erreicht ist, werden die Etiketten aus dem Zwischenspeicher entfernt.

   ANMERKUNG

   Standardwert: 28800 (8 Stunden). Der maximale Wert beträgt 2147483647.

   Die Datei **`product.config`** sollte den folgenden Inhalt haben:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <Common>
        <FileUpdater>
            <PurgeAge>28800</PurgeAge>
        </FileUpdater>
    </Common>
...
</configuration>
```

6. Nach Speichern der Datei wendet der NiceLabel Automation-Dienst die Einstellung automatisch an.

Permanenten Zwischenspeicher aktivieren

PRODUKTEBENEN-INFO

Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

Um den permanenten Zwischenspeicher zu aktivieren und zu konfigurieren, öffnen Sie die Optionen, wählen Sie NiceLabel Automation und aktivieren Sie **Entfernte Dateien zwischenspeichern**.

- **Cachedateien aktualisieren:** Gibt das Zeitintervall (in Minuten) an, nach dem die Dateien im Zwischenspeicher mit den Dateien im ursprünglichen Ordner synchronisiert werden. Dies ist der Zeitraum, über den das System die alte Version der Datei verwenden kann.
- **Cachedateien entfernen, die älter sind als:** Gibt das Zeitintervall (in Tagen) an, nach dem alle Dateien im Zwischenspeicher entfernt werden, auf die innerhalb des festgelegten Zeitraums nicht zugegriffen wurde.

NiceLabel Automation nutzt das folgende lokale Verzeichnis als Zwischenspeicher für entfernte Dateien:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10\FileCache
```

ANMERKUNG

Beim Zwischenspeichern werden Etiketten- und Bilddateiformate unterstützt. Starten Sie nach Aktivierung des Zwischenspeicherns den Automation Dienst neu, um die Änderungen wirksam zu machen.

Neues Laden des Inhalts des Zwischenspeichers erzwingen

NiceLabel Automation Aktualisiert den Inhalte des Zwischenspeichers automatisch nach dem festgelegten Zeitintervall. Der Standardwert sind 5 Minuten.

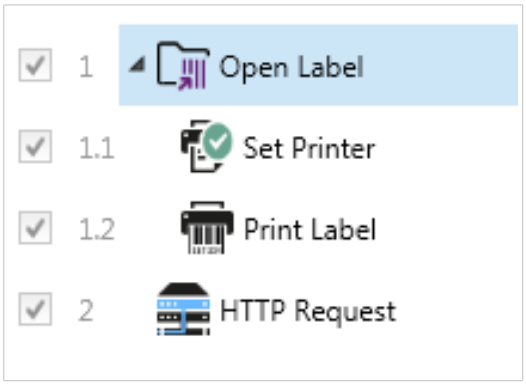
Um ein erneutes Laden des Zwischenspeichers manuell zu erzwingen, tun Sie Folgendes:

1. Öffnen Sie Automation Manager.
2. Suchen Sie die Konfiguration mit dem Trigger, für den Sie die Etiketten neu laden möchten.
3. Klicken Sie mit der rechten Maustaste auf die Konfiguration.
4. Wählen Sie **Konfiguration neu laden**.

7.3. Fehlerhandhabung

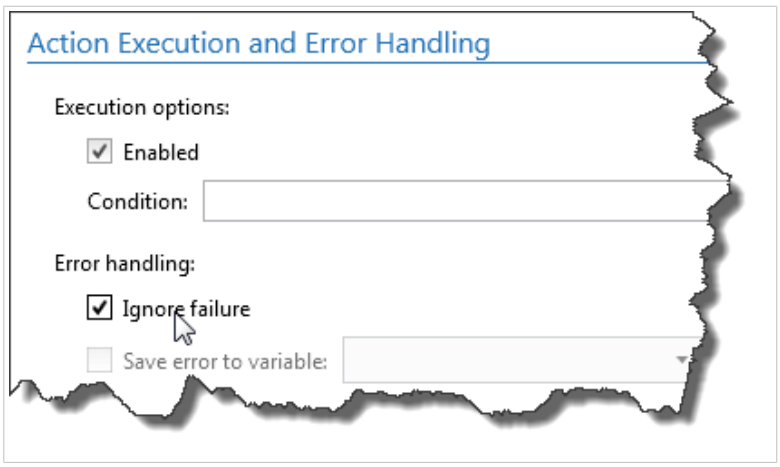
Tritt während der Ausführung einer Aktion ein Fehler auf, beendet NiceLabel Automation die Ausführung aller Aktionen im Trigger. Wenn nach der aktuellen Aktion, die einen Fehler ausgibt, weitere Aktionen definiert sind, werden diese Aktionen nicht ausgeführt.

Ein Beispiel: Die Aktionen sind gemäß dem folgenden Bildschirmfoto definiert. Schlägt die Aktion **Drucker einstellen** wegen eines ungültigen Namens oder eines nicht verfügbaren Druckers fehl, werden die Aktionen **Etikett drucken** und **HTTP-Anfrage** nicht ausgeführt. Die Aktionsverarbeitung bricht bei **Drucker einstellen** ab, Automation Manager zeigt den fehlerhaften Trigger , und das Status-Feedback des Triggers (falls aktiviert) lautet: „falscher Drucker angegeben / Drucker nicht verfügbar“.



In diesem Fall jedoch wollen Sie kein synchrones Feedback verwenden, das automatisch gesendet wird, wenn es für einen Trigger aktiviert wurde, der synchrones Feedback unterstützt. Das Status-Feedback muss asynchron anhand der Aktion **HTTP-Anfrage** bereitgestellt werden, nachdem der Druckauftrag erstellt wurde (bzw. nicht erstellt wurde). Nach Abschluss des Druckprozesses aktualisieren Sie eine Anwendung mit dem Status. Senden Sie zu diesem Zweck eine Nachricht im HTTP-Format an die Anwendung.

In diesem Fall muss die Aktion **HTTP-Anfrage** unabhängig vom Erfolg aller Aktionen erfolgen, die weiter oben in der Liste definiert sind. Aktivieren Sie die Option **Fehler ignorieren** für alle Aktionen über der Aktion **HTTP-Anfrage**. Die Option ist unter der Option **Aktionsausführung und Fehlerbehandlung** einer Aktion verfügbar.



Schlägt eine bestimmte Aktion fehl, beginnt NiceLabel Automation mit der Ausführung der nächsten Aktion in der nächsthöheren Hierarchiestufe.

Beispiel

Wenn die Aktion **Drucker einstellen** auf Ebene 1.1 fehlschlägt, fährt die Ausführung nicht mit der Aktion **Etikett drucken** auf Ebene 1.2 fort, da diese wahrscheinlich ebenfalls fehlschlagen wird. Stattdessen fährt sie mit der Aktion **HTTP-Anfrage** auf Stufe 2 fort, da es sich bei ihr um die nächste Aktion in der nächsthöheren Hierarchiestufe handelt.

Dieselbe Logik kann für die Schleifenausführung von Aktionen verwendet werden, zum Beispiel **Datenfilter verwenden**, **Schleife** und **Für jeden Datensatz**. Mit diesen Aktionen werden alle Elemente in der Liste schrittweise durchgegangen. Falls die Verarbeitung eines der Elemente aus irgendeinem Grund fehlschlägt, stoppt NiceLabel Automation standardmäßig die Verarbeitung aller weiteren Elemente und gibt einen Fehler aus. Wenn Sie die Option **Fehler ignorieren** aktivieren, wird die Verarbeitung des fehlgeschlagenen Elements angehalten, aber NiceLabel Automation fährt mit dem nächsten Element fort. Am Ende wird der Fehler dennoch ausgegeben.

7.4. Synchroner Druckmodus

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

7.4.1. Asynchroner Druckmodus

Der Standard-Betriebsmodus von NiceLabel Automation ist der asynchrone Modus. Der asynchrone Modus ist eine Art von Drucken, bei der ein Trigger Daten für den Druck sendet und die Verbindung zum Druck-Subsystem schließt. Der Trigger wartet nicht auf das Ergebnis des Druckprozesses und empfängt kein Feedback. Direkt nach Senden der Daten ist der Trigger bereit, einen neuen eingehenden Datenstrom anzunehmen.

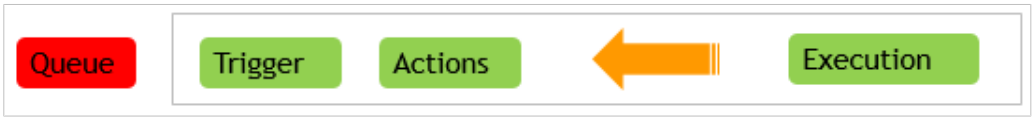
Der asynchrone Modus steigert die Trigger-Performance und die Anzahl von Triggern, die in einem bestimmten Zeitraum verarbeitet werden können. Vor jedem Druckprozess ist eine Pufferzeit vorhanden, in der der Trigger den Druckauftrag einspeist. Der Puffer sorgt dafür, dass während Trigger-Spitzen keine Daten verloren gehen.

Falls bei der Verarbeitung ein Fehler auftritt, wird dieser zwar in Automation Manager (und im NiceLabel Control Center, falls es genutzt wird) protokolliert, aber der Trigger selbst erhält kein entsprechendes Feedback. Bei der Verwendung von Automation im asynchronen Modus können Sie keine von Bedingungen abhängigen Aktionen definieren, die ausgeführt werden, falls bei der Trigger-Ausführung ein Fehler auftritt.

7.4.2. Synchroner Druckmodus

Im Vergleich zum asynchronen Modus wird im synchronen Modus nicht die Verbindung getrennt, wenn der Druckprozess beginnt. In diesem Modus sendet der Trigger Druckdaten und behält die Verbindung zum Druck-Subsystem bei, solange es mit der Ausführung von Aktionen beschäftigt ist. Nach Abschluss des Druckprozesses (erfolgreich oder mit Fehlern) erhält der Trigger Feedback zum Status.

Sie können diese Informationen innerhalb der im selben Trigger definierten Aktionen verwenden und festlegen, dass im Fall eines Fehlers eine andere Aktion ausgeführt werden soll. Außerdem können Sie den Druckauftragsstatus an die datengebende Anwendung zurücksenden. Weitere Informationen finden Sie im Abschnitt Feedback zum Status von Druckaufträgen im Benutzerhandbuch.



Beispiel

Sie können den Druckstatus an die ERP-Anwendung senden, die die Daten bereitgestellt hat.

Sie werden den synchronen Modus nutzen, wenn Sie Status-Feedback innerhalb des Triggers empfangen oder die Nutzung des FIFO-Druckmodus sicherstellen möchten. In diesem Fall werden die Daten, die mit Trigger-Ereignissen empfangen werden, in derselben Reihenfolge wie beim Empfang gedruckt.

ANMERKUNG

Wenn ein Trigger im synchronen Druckmodus ausgeführt wird, kommuniziert er nur mit einem einzelnen Druckprozess. Die Aktivierung des synchronen Druckmodus stellt sicher, dass Daten in der Reihenfolge gedruckt werden, in der sie empfangen wurden (FIFO-Prinzip). Die Mehrkern-Verarbeitung an sich sagt nichts über die Druckreihenfolge aus.

Den synchronen Druckmodus aktivieren

Der Druckmodus kann pro Trigger aktiviert werden. So aktivieren Sie den synchronen Modus in einem Trigger:

1. Öffnen Sie die Eigenschaften des Triggers.
2. Öffnen Sie die Registerkarte **Einstellungen**.
3. Wählen Sie **Andere**.
4. Aktivieren Sie im Bereich **Feedback von der Print Engine** die Option **Überwachtes Drucken**.

7.5. Feedback zum Status von Druckaufträgen

Die Anwendung, die Daten für den Etikettendruck in NiceLabel Automation bereitstellt, erwartet möglicherweise Informationen zu den Status des Druckauftrags. Dieses Feedback kann ein einfaches „Alles OK“ sein, nachdem der Druckauftrag erfolgreich erstellt wurde, oder im Fall von Problemen ausführliche Fehlerbeschreibungen beinhalten. Aus Performance-Gründen ist das Feedback standardmäßig in NiceLabel Automation NiceLabel Automation deaktiviert. So wird ein hoher Druckdurchsatz gewährleistet, da der Trigger sich nicht um die Ausführung von Druckprozessen kümmert. Die Fehler werden in der Protokolldatenbank gespeichert, aber der Trigger ist davon unabhängig.

Sie können diese Methode auch nutzen, um Feedback zu anderen Daten zu senden, die der Trigger sammelt. Dabei kann es sich um den Status der Netzwerkdrucker, die Anzahl von Aufträgen im Drucker-Spooler, eine Liste von Etiketten in einem Ordner, die Liste von Variablen in der angegebenen Etikettendatei und vieles mehr handeln.

ANMERKUNG

Um Feedback-Unterstützung in der Druck-Engine zu aktivieren, aktivieren Sie den synchronen Druckmodus. Weitere Informationen finden Sie im Abschnitt Synchroner Druckmodus.

Aktivieren Sie Feedback zum Status von Druckaufträgen anhand einer der beiden verfügbaren Methoden.

Der Trigger stellt Feedback zum Status des Druckauftrags bereit (synchrones Feedback)
Einige Trigger haben eine integrierte Feedback-Funktion. Wenn der synchrone Druckmodus aktiviert ist, wird der Status des Druckauftrags automatisch an den Trigger ausgegeben. Der Client kann die Daten an den Trigger senden, die Verbindung aufrechterhalten und auf das Feedback warten. Um diese Feedback-Methode zu verwenden, wählen Sie einen Trigger-Typ aus, der die Bereitstellung von Feedback unterstützt.

Tritt bei einer der Aktionen ein Fehler auf, enthält die interne Variable **`ActionLastErrorDesc`** die entsprechende ausführliche Fehlermeldung. Sie können ihren Wert 1:1 senden oder ihn anpassen.

Weitere Informationen finden Sie in den Angaben zu den jeweiligen Trigger-Typen.

- Webdienst-Trigger: Dieser Trigger-Typ verfügt über eine integrierte Unterstützung für Feedback. Das WSDL-(Web Service Description Language-)Dokument enthält Angaben zur Webdienst-Schnittstelle und zur Aktivierung von Feedback. Sie können die Standardantwort nutzen, die eine Fehlerbeschreibung ausgibt, falls die Druckaktion fehlschlägt. Alternativ können Sie die Nachricht auch anpassen und den Inhalt einer beliebigen Variablen senden. Die Variable selbst kann beliebige Daten enthalten, einschließlich einer Etikettenvorschau oder eines Etiketten-Druckauftrags (Binärdaten).
- HTTP Server Trigger: Dieser Trigger-Typ verfügt über eine integrierte Unterstützung für Feedback. NiceLabel Automation nutzt die Standard-HTTP-Antwortcodes, um den Status des Druckauftrags anzuzeigen. Sie können die HTTP-Antwort auch anpassen und den Inhalt einer beliebigen Variablen senden. Die Variable selbst kann beliebige Daten enthalten, einschließlich einer Etikettenvorschau oder eines Etiketten-Druckauftrags (Binärdaten).
- TCP/IP Server Trigger: Dieser Trigger unterstützt Feedback, aber nicht automatisch. Damit er Feedback ausgeben kann, konfigurieren Sie den datengebenden Client dafür, die Verbindung nicht zu trennen, nachdem die Daten gesendet wurden. Nach Abschluss des Druckprozesses ist für die nächste Aktion in der Liste (Daten an TCP/IP Port senden) möglicherweise die Einstellung Absender antworten sein aktiviert. Senden Sie Feedback über die bestehende, noch offene Verbindung.

Aktion stellt Feedback zum Status des Druckauftrags bereit (asynchrones Feedback)
Wenn Sie Trigger verwenden, die keine native Unterstützung für Feedback bieten, oder wenn Sie Feedback-Nachrichten während der Trigger-Verarbeitung senden möchten, definieren Sie eine Aktion, die Feedback an ein bestimmtes Ziel sendet. In diesem Fall kann die datengebende Anwendung die Verbindung trennen, nachdem die Trigger-Daten bereitgestellt wurden.

Beispiel

Sie haben den TCP/IP-Trigger verwendet, um die Daten zu erfassen. Der Client hat die Verbindung direkt nach Senden der Daten unterbrochen, weswegen keine Antwort über diese Verbindung mehr möglich ist. In solchen Fällen können Sie einen anderen Kanal verwenden, um Feedback zu senden. Sie können eine der Aktionen für ausgehende Verbindungen konfigurieren, z. B. SQL-Anweisung ausführen, Dokument/Programm öffnen, HTTP-Anfrage, Daten an TCP/IP-Port senden und andere. Solche Aktionen werden unter der Aktion Etikett drucken platziert.

Wenn Sie nur zu bestimmten Status Feedback senden möchten, etwa zu „Fehler aufgetreten“, können Sie eine der folgenden Methoden verwenden.

- **Bedingung für eine Aktion nutzen:** Der Druckauftrags-Status wird in Form von zwei internen Variablen angegeben (**`ActionLastErrorID`** und **`ActionLastErrorDesc`**). Die erste enthält die Fehler-ID oder, falls keine Fehler aufgetreten sind, den Wert **`0`**. Die zweite enthält eine ausführliche Fehlermeldung. Verwenden Sie Werte dieser Variablen in Bedingungen für Aktionen, die im Fall von Fehlern ausgeführt werden sollen. Beispielsweise würden Sie die Aktion **HTTP-Anfrage** nach dem Drucken verwenden. Die Aktion würde im Fall eines Fehlers Feedback senden. Um solches Feedback zu aktivieren, tun Sie Folgendes:

  1. Öffnen Sie die Trigger-Eigenschaften.
  2. Klicken Sie in der Multifunktionsleisten-Gruppe **Variable** auf die Schaltfläche **Interne Variablen** und aktivieren Sie die Variable **`ActionLastErrorID`**.
  3. Öffnen Sie die Registerkarte „Aktionen“.
  4. Fügen Sie die Aktion **Daten an HTTP senden** hinzu.
  5. Erweitern Sie innerhalb der Eigenschaften der Aktion den Punkt **Optionen für Ausführung und Fehlerhandhabung anzeigen**.
  6. Geben Sie Folgendes für **Bedingung** ein. Die Aktion mit dieser Bedingung wird nur ausgeführt, wenn ein Fehler auftritt und **`ActionLastErrorID`** die Fehler-ID enthält (ein beliebiger Wert größer **`0`**). Standardmäßig werden die Bedingungen mithilfe der VBScript-Syntax angegeben.

     ```
     ActionLastErrorID > 0
     ```

  7. Außerdem müssen Sie die Option **Fehler ignorieren** für jede Aktion öffnen, die erwartungsgemäß fehlschlagen wird. So weisen Sie an, die Ausführung der Aktionen nicht vollständig zu beenden, sondern mit der nächsten Aktion auf derselben hierarchischen Ebene fortzufahren.

> **ANMERKUNG**
> Weitere Informationen finden Sie in den Abschnitten Fehlerhandhabung.

- **Aktion „Testen“ verwenden:** Die Aktion „Testen“ macht die Programmierung von Bedingungen überflüssig. Die Aktion bietet Ihnen zwei Platzhalter. Der Platzhalter **Ausführen** enthält Aktionen, die Sie ausführen möchten. Falls bei ihrer Ausführung ein Fehler passiert, werden sie abgebrochen; stattdessen werden die Aktionen ausgeführt, die im Platzhalter **Bei Fehler** angegeben sind. In diesem Platzhalter können Sie Aktionen für ausgehende Verbindungen verwenden, um ein Status-Feedback zum Druckauftrag bereitzustellen. Weitere Informationen finden Sie im Abschnitt Testen.

7.6. Drucker vom automatisierten Drucken ausschließen

Unter bestimmten Umständen müssen Sie in Ihrer Druckumgebung einige Ihrer Drucker vom automatisierten Druckprozess ausschließen. Mögliche Gründe, weswegen Sie Drucker vom automatisierten Drucken ausschließen sollten, sind die Druckrichtlinien Ihres Unternehmens oder die Einschränkungen im Rahmen Ihrer Lizenz.

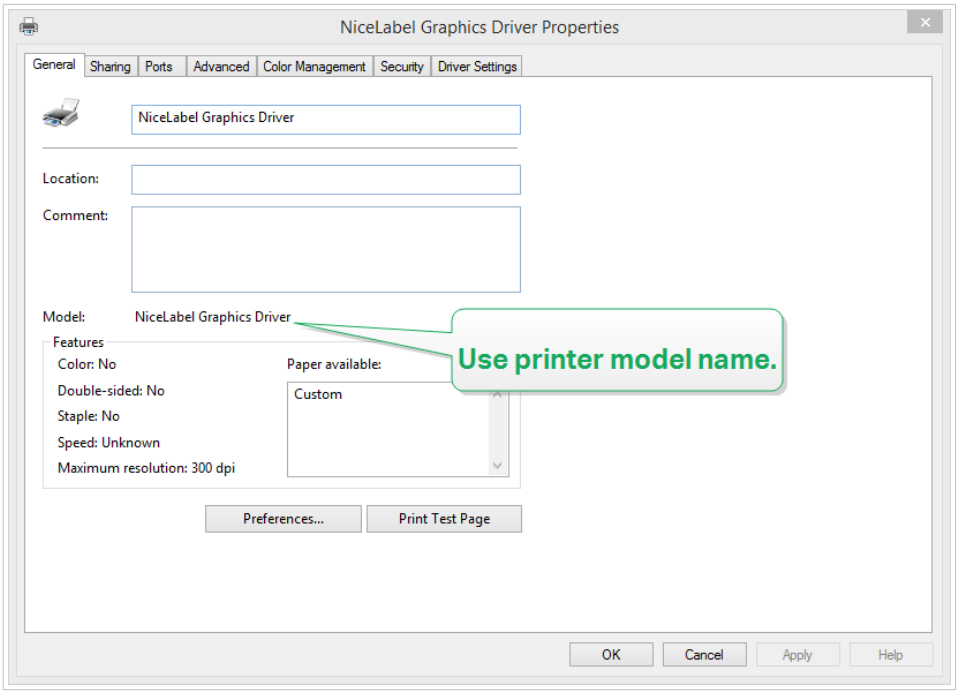
Standardmäßig verhindert Automation das automatisierte Drucken mithilfe von Dateidruckern wie Microsoft Shared Fax Driver, Microsoft Print To PDF, Microsoft XPS Document Writer und ähnlichen. Diese Dateidrucker erfordern, dass Benutzer den Speicherort für ihre „Ausdrucke“ manuell festlegen. Die Anforderung eines manuellen Benutzereingriffs führt dazu, dass die Druck-Engine anhält und einen Fehler ausgibt.

So verhindern Sie, dass Automation bestimmte Drucker in den laufenden Konfigurationen nutzt:

> **ANMERKUNG**
> Wenn Sie die vom automatisierten Drucken ausgeschlossenen Drucker in der Datei **`product.config`** angeben, müssen Sie auch Ihre Dateidrucker explizit auflisten.

1. Öffnen Sie die Datei **`product.config`** in einem Text-Editor.
   Die Datei befindet sich hier:
   `%PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config`
2. Erstellen Sie eine Sicherungskopie der Datei **`product.config`**.
3. Automation Verwendet zwei Parameter, um zu prüfen, welches Druckermodell und welche Druckerschnittstelle vom automatisierten Drucken ausgeschlossen werden sollten. Fügen Sie die folgenden Parameter zu Ihrer Datei **`product.config`** hinzu.
   - **`/IntegrationService/DisabledPrinterDrivers`** und geben Sie die Druckermodelle ein, die Sie von automatisierten Drucken ausschließen möchten.

- **`/IntegrationService/DisabledPrinterPorts`** und geben Sie die Schnittstellen ein, die Sie von automatisierten Drucken ausschließen möchten.




```
<configuration>
    <IntegrationService>
        <DisabledPrinterDrivers>NiceLabel Graphics
Driver,NicePrinter1200dpi</DisabledPrinterDrivers>
        <DisabledPrinterPorts>LPT1,LPT2</DisabledPrinterPorts>
    </IntegrationService>
</configuration>
```

4. Nach Festlegen der Druckermodelle und -schnittstellen in der Datei **`product.config`** können Sie Ihre Automation Konfigurationen weiterhin ausführen, aber Ihre aktualisierten Einstellungen verhindern das Drucken auf den aufgelisteten Druckern.Automation meldet einen Fehler, wenn diese Drucker Teil der ausgeführten Konfigurationen sind.

> **ANMERKUNG**
> Das automatisierte Drucken wird nicht gestoppt, wenn die ausgeschlossenen Drucker in der Aktion Druck an Datei umleiten verwendet werden.

7.7. Speichern/Abrufen-Druckmodus verwenden

Der Speichern/Abrufen-Modus optimiert den Druckprozess. Er beschleunigt die Reaktion des Druckers, indem er die Datenmenge reduziert, die im Rahmen von sich wiederholenden Druckaufgaben an den Drucker gesendet wird.

Wenn der Speichern/Abrufen-Modus aktiviert ist, muss NiceLabel Automation nicht für jeden Ausdruck die vollständigen Etikettendaten senden. Stattdessen werden die Etiketten (Vorlagen) im Druckerspeicher abgelegt. Feste Objekte werden als solche gespeichert, für variable Objekte werden Platzhalter definiert.NiceLabel Automation sendet nur Daten für variable Etikettenobjekte und die Abrufbefehle. Der Drucker wendet die empfangenen Daten in den Platzhaltern auf dem gespeicherten Etikett an und druckt das Etikett per Abruf aus dem internen Speicher aus. Für gewöhnlich werden nur einige Bytes an Daten an den Drucker gesendet (beim normalen Druck wäre es einige Kilobyte).

Die Aktion besteht aus zwei Prozessen:

- **Etikett speichern:** Während dieses Prozesses erstellt die Anwendung eine Beschreibung der Etikettenvorlage, die in der ausgewählten Druckerbefehlssprache formatiert ist. Danach sendet die Anwendung die erstellte Befehlsdatei an den Druckerspeicher, wo sie abgelegt wird. Sie können ein Etikett aus dem Etikettendesigner heraus oder NiceLabel Automation anhand der Aktion Etikett im Drucker speichern.

  > **ANMERKUNG**
  > In den Eigenschaften des Etiketts muss der Speichern/Abrufen-Modus definiert sein, bevor Sie es im Druckerspeicher ablegen können.

- **Etikett abrufen (drucken):** Ein im Druckerspeicher abgelegtes Etikett wird umgehend gedruckt. Anhand des Abrufen-Prozesses erstellt NiceLabel Automation eine weitere Befehlsdatei, um den Drucker anzuweisen, welches Etikett aus seinem Speicher gedruckt werden soll. Die tatsächliche an den Drucker gesendete Datenmenge hängt von der jeweiligen Situation ab. Für feste Etiketten ohne variable Inhalte enthält die Abrufen-Befehlsdatei nur den Abrufbefehl für das Etikett. Für Etiketten mit variablen Feldern enthält die Befehlsdatei die Werte für die enthaltenen Variablen sowie den Abrufbefehl für das Etikett.
  Um ein Etikett aus NiceLabel Automation abzurufen, verwenden Sie einfach eine der üblichen Druckaktionen. Nach der Ausführung analysiert die Aktion das Etikett und aktiviert den entsprechenden Druckmodus: normaler Druck oder Abrufdruck, je nach Definition im Etikett.

> **WARNUNG**
> Vor Aktivierung dieses Modus sollten Sie sicherstellen, dass der entsprechende Druckertreiber für den Etikettendrucker ausgewählt ist. Nicht alle Etikettendrucker unterstützen den Speichern/Abrufen-Druckmodus. Der Druckertreiber, für den das Etikett im Etikettendesigner erstellt wurde, muss zudem auf dem Rechner installiert sein, auf dem NiceLabel Automation ausgeführt wird.

7.8. Hochverfügbarkeits-(Failover-)Cluster

PRODUKTEBENEN-INFO
Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

NiceLabel Automation unterstützt Microsoft Hochverfügbarkeits-(Failover-)Cluster. Ein Failover-Cluster ist eine Gruppe unabhängiger Computer, die zusammenarbeiten, um die Verfügbarkeit des Etikettendrucks über NiceLabel Automation zu steigern. Die Server im Cluster (Knoten genannt) werden durch physische Kabel und Software miteinander verbunden. Falls ein Knoten (oder mehrere Knoten) im Cluster ausfällt, stellen andere Knoten den jeweiligen Dienst bereit (dieses Verfahren nennt man Failover). Zudem werden die Cluster-Dienste proaktiv überwacht, um sicherzustellen, dass sie einwandfrei funktionieren. Ist dies nicht der Fall, werden sie neu gestartet oder auf einen anderen Knoten verlegt. Die Clients, die Daten bereitstellen, verbinden sich mit der IP-Adresse, die zu dem gesamten Cluster gehört, und nicht mit den IP-Adressen einzelner Knoten.

Um die Ausführung von NiceLabel Automation in einem Hochverfügbarkeits-Cluster zu ermöglichen, tun Sie Folgendes:

- Richten Sie die Funktion „Microsoft Failover Clustering“ auf Ihren Windows-Servern ein.
- Installieren Sie NiceLabel Automation auf jedem Knoten.
- Aktivieren Sie die Failover-Cluster-Unterstützung in den NiceLabel Automation-Einstellungen auf jedem Knoten.
  Tun Sie Folgendes:
  1. Öffnen Sie **Datei** > **Optionen** > **Automation**.
  2. Aktivieren Sie unter der Gruppe **Cluster-Unterstützung** die Option **Unterstützung für Failover-Cluster**.
  3. Navigieren Sie zu einem Ordner, der sich außerhalb der beiden Knoten befindet, auf den die NiceLabel Automation Software aber dennoch mit vollen Zugriffsberechtigungen zugreifen kann. Wichtige Systemdateien, die beide Knoten benötigen, werden in diesen Ordner kopiert.
- Konfigurieren Sie den Cluster so, dass NiceLabel Automation auf dem zweiten Knoten gestartet wird, falls der Master-Knoten ausfällt.

7.9. Lastausgleichs-Cluster

PRODUKTEBENEN-INFO
Der in diesem Abschnitt beschriebene Funktionsumfang steht in LMS Enterprise zur Verfügung.

NiceLabel Automation bietet Unterstützung für Microsoft Lastausgleichs-Cluster. Ein Lastausgleichs-Cluster ist eine Gruppe unabhängiger Computer, die zusammenarbeiten, um die Verfügbarkeit und Skalierbarkeit des Etikettendrucks über NiceLabel Automation zu steigern. Die Server im Cluster (Knoten genannt) werden durch physische Kabel und Software miteinander verbunden. Die eingehenden Etikettendruck-Anfragen werden unter allen Knoten in einem Cluster aufgeteilt. Die Clients, die Daten bereitstellen, verbinden sich mit der IP-Adresse, die zu dem Cluster gehört, und nicht mit den IP-Adressen einzelner Knoten.

ANMERKUNG

Sie können TCP/IP-basierte Trigger mit Lastausgleichs-Clustern verwenden. Diese Trigger sind TCP/IP Server Trigger, HTTP Server Trigger, Webdienst-Trigger und Cloud-Trigger.

So aktivieren Sie NiceLabel Automation für den Lastausgleich:

- Richten Sie die Funktion „Microsoft Load-balancing Clustering“ auf Ihren Windows-Servern ein.
- Installieren Sie NiceLabel Automation auf jedem Knoten.
- Laden Sie die gleichen Konfigurationsdateien in Automation Managerjeden Knoten.

8. Informationen zu Datenstrukturen

Dieser Abschnitt beschäftigt sich mit Datenstrukturen, die in Automationsszenarios häufig zum Einsatz kommen. Wenn wir mit verschiedenen Datendateien arbeiten, müssen wir ihre Struktur analysieren, die relevanten Werte aus den gewünschten Feldern extrahieren und sie auf Etiketten drucken. Jeder der im Folgenden aufgeführten Fälle wird in Beispielkonfigurationen verwendet, die im Automation-Installationspaket enthalten sind.

- Textdatenbank
- Zusammengesetzte CSV-Dateien
- Binärdateien
- Altdaten
- Befehlsdateien
- XML-Daten
- JSON Data

8.1. Binärdateien

Binärdateien enthalten reinen Text und Binärzeichen wie Steuercodes (Zeichen unter ASCII-Code 32). Der Filter für unstrukturierte Daten unterstützt binäre Zeichen. Sie können Binärzeichen verwenden, um Feldpositionen zu definieren und Feldwerte anzugeben.

Ein typisches Beispiel wäre ein Datenexport aus einem Altsystem, in dem die Daten für jedes Etikett mithilfe eines Seitenvorschub-Zeichens **<FF>** getrennt werden.

8.1.1. Beispiel

In diesem Fall erfasst der Automation-Trigger den Druckdatenstrom. Der gelb hervorgehobene Datenbereich muss aus dem Datenstrom extrahiert und an einen anderen Drucker gesendet werden. Der Filter ist dafür konfiguriert, nach **<FF>** als Feldabschluss zu suchen.

```
<ESC>%-12345X@PJL USTATUSOFF
@PJL INFO STATUS
@PJL USTATUS DEVICE=ON
<ESC>%-12345X<ESC>%-12345X



^^02^L
^^02^O0270
D11
H15
PE
SE
Q0001
131100000300070001-001-001
1e4205500750050000100 1019
1322000001502859
W
E
<FF><ESC>%-12345X<ESC>%-12345X@PJL USTATUSOFF
<ESC>%-12345X
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

8.2. Befehlsdateien

Befehlsdateien sind reine Textdateien mit Befehlen, die nacheinander von oben nach unten ausgeführt werden.NiceLabel Automation unterstützt native Befehlsdateien sowie Oracle- und SAP XML-Befehlsdateien. Weitere Informationen finden Sie im Abschnitt Spezifikationen für Befehlsdateien, Oracle XML-Spezifikationen, und SAP AII XML-Spezifikationen.

8.2.1. Beispiel

Das Etikett **`label2.nlbl`** wird auf dem Drucker **`CAB A3 203DPI`** gedruckt.

```
LABEL "label2.nlbl"
SET code="12345"
SET article="FUSILLI"
SET ean="383860026501"
SET weight="1,0 kg"
PRINTER "CAB A3 203DPI"
PRINT 1
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

8.3. Zusammengesetzte CSV-Dateien

Zusammengesetzte CSV-Dateien sind Textdateien, die Daten in zwei Strukturen bereitstellen: in einer Standard-CSV-Struktur und in einem mehrzeiligen Header, der eine nicht-standardmäßige Struktur nutzt. Der Inhalt von zusammengesetzten CSV-Dateien kann nicht anhand eines einzelnen Filters geparst werden. Um Daten in beiden Strukturen zu parsen, konfigurieren Sie zwei separate Filter:

- Filter für strukturierten Text für Felder mit CSV-Struktur
- Filter für unstrukturierte Daten für Felder im Header mit nicht-standardmäßiger Struktur

Eine Konfiguration, die eine zusammengesetzte CSV-Datei enthält, erfordert zwei Aktionen, damit beide Filter auf die empfangenen Daten angewandt werden.

8.3.1. Beispiel

Die Datenelemente von Zeile 3 bis zum Ende des Dokuments haben eine CSV-Struktur und werden vom Filter für strukturierten Text geparst. Die Datenelemente in den ersten beiden Zeilen haben keine bestimmte Struktur und werden vom Filter für unstrukturierte Daten geparst.

```
OPTPEPPQPF0 NL004002 ;F75-TEP77319022891-001-001
OPT2 zg2lbprt.p 34.1.7.7 GOLF+ label
print"printer";"label";"lbl_qty";"f_logo";"f_field_1";"f_field_2";"f_field_3"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1161P";"Post: ";"1"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1162P";"Post: ";"2"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1163P";"Post: ";"3"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1164P";"Post: ";"4"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1165P";"Post: ";"5"
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

8.4. Altdaten

Altdaten bezeichnet einen unstrukturierten oder teilweise strukturierten Export aus Altanwendungen. Solche Exporte verwenden weder die CSV- noch die XML-Datenstruktur. Um relevante Daten aus solchen Dateien zu extrahieren, verwenden Sie den Filter für unstrukturierte Daten und definieren die Positionen der gewünschten Felder. Der Filter in Automation extrahiert Feldwerte und stellt sie für den Druck auf Etiketten bereit.

8.4.1. Beispiel

Die folgende Datei basiert nicht auf strukturbezogenen Regeln. Jedes Feld muss manuell konfiguriert werden.

```
HAWLEY      ANNIE      ER12345678 ABC      XYZ
                9876543210
PRE OP      07/11/12      F 27/06/47      St. Ken Hospital      3

G015 134 557 564 9      A- 08/11/12      LDBS F-          PB      1
G015 134 654 234 0      A- 08/11/12      LDBS F-          PB      2
G015 134 324 563 C      A- 08/11/12      LDBS F-          PB      3



                Antibody Screen: Negative
    Store Sample :
    SAMPLE VALID FOR 24 HOURS, NO TRANSFUSION HISTORY SUPPLIED

07/11/12      B,31.0001245.E      O Rh(D) Pos          PHO
                        RLUH      BT
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

8.5. Textdatenbank

Textdatenbank ist ein anderer Begriff für eine Textdatei mit strukturierten Feldern, z. B. CSV (Datei mit durch Kommas getrennten Werten) oder eine Textdatei mit Feldern mit fester Spaltenbreite. In beiden Fällen können Sie auf die Schaltfläche **Datenstruktur importieren** klicken und den Schritten im Assistenten folgen, um die Felder zu importieren. Wenn Sie eine Datendatei mit getrennter Struktur vorliegen haben und die Anzahl von Feldern von Kopie zu Kopie variiert, können Sie die Option **Dynamische Struktur** aktivieren und NiceLabel Automation die Datenextraktion und die Zuordnung zu Variablen automatisch ausführen lassen. Weitere Informationen finden Sie im Abschnitt Dynamische Struktur aktivieren.

8.5.1. Beispiel

- **Datei mit getrennten Feldern:** Die erste Zeile der Datei enthält Feldnamen, die der Filter importieren kann.

```
Product_ID;Code_EAN;Product_desc;Package
CAS006;8021228110014;CASONCELLI ALLA CARNE 250G;6
```

```
PAS501;8021228310001;BIGOLI 250G;6
PAS502GI;8021228310018;TAGLIATELLE 250G;6
PAS503GI;8021228310025;TAGLIOLINI 250G;6
PAS504;8021228310032;CAPELLI D'ANGELO 250G;6
```

- **Datei mit Feldern mit fester Spaltenbreite:** Die Felder enthalten eine feste Anzahl von Zeichen.

```
CAS006     8021228110014 CASONCELLI ALLA CARNE 250G 6
PAS501     8021228310001 BIGOLI 250G                6
PAS502GI   8021228310018 TAGLIATELLE 250G           6
PAS503GI   8021228310025 TAGLIOLINI 250G            6
PAS504     8021228310032 CAPELLI D'ANGELO 250G      6
```

Weitere Informationen finden Sie in den Abschnitten Beispiele.

8.6. XML-Daten

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

XML steht für eXtensible Markup Language. XML-Tags sind nicht vordefiniert; Sie können Ihre eigenen Tags zur Beschreibung Ihrer Daten definieren. XML wurde als selbsterklärende Sprache entwickelt.

Die XML-Struktur wird durch Elemente, Attribute (und deren Werte) und Text (Element-Text) definiert.

8.6.1. Beispiele

Oracle XML

Die Verarbeitung von Oracle XML ist ein grundlegender Bestandteil der Software. Sie müssen keine Filter zur Extraktion der Daten definieren, sondern nur die integrierte Aktion Oracle XML-Befehlsdatei ausführen. Weitere Informationen zur XML-Struktur finden Sie im Abschnitt Oracle XML-Spezifikationen.

```
<?xml version="1.0" standalone="no"?>
<labels _FORMAT="case.nlbl" _PRINTERNAME="Production01" _QUANTITY="1">
    <label>
        <variable name="CASEID">0000000123</variable>
        <variable name="CARTONTYPE"/>
        <variable name="ORDERKEY">0000000534</variable>
        <variable name="BUYERPO"/>
        <variable name="ROUTE"></variable>
        <variable name="CONTAINERDETAILID">0000004212</variable>
        <variable name="SERIALREFERENCE">0</variable>
        <variable name="FILTERVALUE">0</variable>
        <variable name="INDICATORDIGIT">0</variable>
        <variable name="DATE">11/19/2012 10:59:03</variable>
    </label>
</labels>
```

Allgemeines XML

Wird die XML-Struktur in der Software nicht nativ unterstützt, müssen Sie den XML-Filter sowie Regeln für die Datenextraktion definieren. Weitere Informationen finden Sie im Abschnitt Informationen zu Datenstrukturen.

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
    <asx:values>
        <NICELABEL_JOB>
            <TIMESTAMP>20130221100527.788134</TIMESTAMP>
            <USER>PGRI</USER>
            <IT_LABEL_DATA>
                <LBL_NAME>goods_receipt.nlbl</LBL_NAME>
                <LBL_PRINTER>Production01</LBL_PRINTER>
                <LBL_QUANTITY>1</LBL_QUANTITY>
                <MAKTX>MASS ONE</MAKTX>
                <MATNR>28345</MATNR>
                <MEINS>KG</MEINS>
                <WDATU>19.01.2012</WDATU>
                <QUANTITY>1</QUANTITY>
                <EXIDV>012345678901234560</EXIDV>
            </IT_LABEL_DATA>
        </NICELABEL_JOB>
    </asx:values>
</asx:abap>
```

NiceLabel-XML

Die Verarbeitung von NiceLabel XML ist ein grundlegender Bestandteil der Software. Sie müssen keine Filter zur Extraktion der Daten definieren, sondern nur die integrierte Aktion Befehlsdatei ausführen. Weitere Informationen zur XML-Struktur finden Sie im Abschnitt XML-Befehlsdatei.

```
<nice_commands>
    <label name="label1.nlbl">

        <session_print_job printer="CAB A3 203DPI" skip=0 job_name="job name
1" print_to_file="filename 1">
            <session quantity="10">
                <variable name="variable name 1" >variable value 1</variable>
            </session>
        </session_print_job>

        <print_job printer="Zebra R-402” quantity="10" skip=0
identical_copies=1 number_of_sets=1 job_name="job name 2"
print_to_file="filename 2">
            <variable name="variable1" >1</variable>
            <variable name="variable2" >2</variable>
            <variable name="variable3" >3</variable>
        </print_job>
    </label>
</nice_commands>
```

Weitere praktische Informationen zur Arbeit mit XML-Daten finden Sie im Abschnitt Beispiele.

8.7. JSON-Daten

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

JavaScript Object Notation (JSON) ist ein Open-Standard-Dateiformat. JSON nutzt menschenlesbaren Text zur Übermittlung von Datenobjekten, die aus Namen-Wert-Paaren bestehen, und von Datenfeld-Datentypen (oder einem anderen serialisierbaren Wert). JSON ist ein sehr häufiges Datenformat für asynchrone Browser-Server-Kommunikation und wird unter anderem als Ersatz für XML verwendet.

Es gibt mehrere Online-Ressourcen, die die Ähnlichkeiten und Unterschiede zwischen JSON und XML beschreiben. In der folgenden Tabelle werden einige davon beschrieben:

JSON	XML
Ist JavaScript Object Notation	Ist Extensible Markup Language
Basiert auf JavaScript-Sprache.	Ist von SGML abgeleitet.
Ist eine Methode zur Darstellung von Objekten.	Ist eine Markup-Sprache und nutzt eine Tagstruktur, um Datenelemente darzustellen.
Bietet keine Unterstützung für Namespaces.	Unterstützt Namespaces.
Unterstützt Datenfelder.	Unterstützt keine Datenfelder.
Dateien sind im Vergleich zu XML sehr leicht lesbar.	Dokumente sind vergleichsweise schwer lesbar und interpretierbar.
Nutzt kein End-Tag.	Hat Start- und End-Tags.
Ist weniger sicher.	Ist sicherer als JSON.
Unterstützt keine Kommentare.	Unterstützt Kommentare.
Unterstützt nur UTF-8-Codierung.	Unterstützt verschiedene Codierungen.

Quelle: https://www.geeksforgeeks.org/difference-between-json-and-xml/

Beispiele

```
{
  "DELIVERYNOTE": {
    "LIST_CUSTOMER_INFO": {
      "CUSTOMER_INFO": {
        "CUSTOMER_NAME": "Customer A",
        "CUSTOMER_STREET_ADDRESS": "Test St",
        "CUSTOMER_POST_ADDRESS": "1234, Test City",
        "CUSTOMER_NUMBER": "1234",
        "CURRENCY": "EUR",
        "DELIVERY_METHOD": "Express delivery",
        "EDI_INFORMATION": "EDI",
        "ORDER_TYPE": "CSO",
        "ORDER_NUMBER": "123",
        "LIST_ITEM": {
          "ITEM": [
            {
              "ARTICLE_NUMBER": "0001",
              "ARTICLE_NAME": "Collins Complete Woodworker's Manual",
              "PRICE": "23.3"
            },
            {
              "ARTICLE_NUMBER": "0002",
              "ARTICLE_NAME": "Be Careful What You Wish For (Clifton
Chronicles)",
              "PRICE": "16.6"
            },
            {
              "ARTICLE_NUMBER": "0003",
              "ARTICLE_NAME": "Mockingjay (part III of Hunger Games Trilogy)",
              "PRICE": "25.0"
            }
          ]
        }
      }
    }
  }
}
```

```
{
    "NICELABEL_JOB": {
      "TIMESTAMP": "20130221100527.788134",
      "USER": "PGRI",
       IT_LABEL_DATA": {
        "LBL_NAME": "goods_receipt.nlbl",
        "LBL_PRINTER": "Production01",
```

```
        "LBL_QUANTITY": "1",
        "MAKTX": "MASS ONE",
        "MATNR": "28345",
        "MEINS": "KG",
        "WDATU": "19.01.2012",
        "QUANTITY": "1",
        "EXIDV": "012345678901234560"
      }
    }
  }
}
```

9. Referenz und Fehlerbehebung

9.1. Typen von Befehlsdateien

9.1.1. Spezifikationen für Befehlsdateien

Befehlsdateien enthalten Anweisungen für den Druckprozess. Diese Anweisungen werden anhand von NiceLabel Befehlen ausgedrückt. Befehle werden hintereinander ausgeführt, vom Anfang bin zum Ende der Datei. Die Dateien unterstützen Unicode-Formatierung, sodass Sie mehrsprachige Inhalte einschließen können. Es gibt drei verschiedene Arten von Befehlsdateien.

9.1.2. CSV-Befehlsdatei

Die in CSV-Befehlsdateien verfügbaren Befehle sind eine Untermenge der NiceLabel Befehle. Die folgenden Befehle sind verfügbar: **LABEL**, **SET**, **PORT**, **PRINTER** und **PRINT**.

CSV steht für Comma Separated Values (durch Kommas getrennte Werte). CSV-Dateien sind Textdateien, deren Werte durch Kommas (`,`) getrennt werden. Solche Textdateien können Unicode-Werte enthalten (wichtig für mehrsprachige Daten). Jede Zeile in der CSV-Befehlsdatei enthält Befehle für eine einzelne Etikettendruckaktion.

Die erste Zeile muss Befehle und Variablennamen enthalten. Die Reihenfolge der Befehle und Namen ist nicht wichtig, aber alle Datensätze im selben Datenstrom müssen dieselbe Struktur aufweisen. Variable **`Name-Wert`**-Paare werden automatisch extrahiert und an das referenzierte Etikett gesendet. Falls eine Variable mit einem Namen aus der CSV-Datei auf dem Etikett nicht vorhanden ist, wird keine Fehlermeldung ausgegeben.

9.1.2.1. Beispiel für eine CSV-Befehlsdatei

Das Beispiel zeigt eine strukturelle Ansicht der Felder, die Sie in einer CSV-Datei verwenden können.

```
@Label,@Printer,@Quantity,@Skip,@IdenticalCopies,NumberOfSets,@Port,Product_ID,
 Product_Name
label1.nlbl, CAB A3 203 DPI, 100, , , , , 100A, Product 1
label2.nlbl, Zebra R-402, 20, , , , , 200A, Product 2
```

Spezifikation der CSV-Befehle

Die Befehle in der ersten Datenzeile müssen mit dem at-Zeichen (@) ausgedrückt werden. Die Felder ohne @ am Anfang sind Namen von Variablen. Diese Felder werden zusammen mit ihren Werten als **`Name-Wert`**-Paare extrahiert.

- **@Label:** Gibt den zu verwendenden Etikettennamen an. Es empfiehlt sich, den Pfad und Dateinamen des Etiketts anzugeben. Stellen Sie sicher, dass der Dienstbenutzer auf die Datei zugreifen kann. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk im NiceLabel Automation Benutzerhandbuch. Ein erforderliches Feld.
- **@Printer:** Gibt den zu verwendenden Drucker an. Der Befehl übergeht den in der Etikettenvorlage definierten Drucker. Stellen Sie sicher, dass der Dienstbenutzer auf den jeweiligen Drucker zugreifen kann. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk. Optionales Feld.
- **@Quantity:** Gibt die Anzahl von zu druckenden Etiketten an. Mögliche Werte: numerischer Wert, VARIABLE oder UNLIMITED. Weitere Informationen finden Sie im Abschnitt Etikett drucken. Ein erforderliches Feld.
- **@Skip:** Gibt die Anzahl von Etiketten an, die auf der ersten gedruckten Seite übersprungen werden sollen. Diese Funktion ist nützlich, wenn Sie eine teilweise bedruckte Seite erneut verwenden möchten. Optionales Feld.
- **@IdenticalCopies:** Gibt die Anzahl von Kopien an, die für das jeweilige Etikett gedruckt werden sollen. Diese Funktion ist nützlich, wenn Sie Etiketten mit Daten aus einer Datenbank drucken oder wenn Sie Zähler verwenden und Kopien von Etiketten benötigen. Optionales Feld.
- **@NumberOfSets:** Gibt an, wie viele Male der Druckprozess wiederholt werden soll. Jedes Etiketten-Set entspricht einer einzelnen Instanz des Druckprozesses. Optionales Feld.
- **@Port:** Gibt den Namen der Schnittstelle für den Drucker an. So können Sie die im Druckertreiber festgelegte Schnittstelle umgehen. Sie können den Befehl auch nutzen, um den Druck an eine Datei umzuleiten. Optionales Feld.
- **Andere Feldnamen:** Alle anderen Felder definieren die Namen von Variablen auf dem Etikett. Die Inhalte der Felder werden in der Variablen gespeichert, die denselben Namen hat wie ihr Wert.

9.1.3. JOB-Befehlsdatei

JOB-Befehlsdateien sind Textdateien mit NiceLabel Befehlen. Die Befehle werden von oben nach unten ausgeführt. Die Befehlskette beginnt meistens mit LABEL (um ein Etikett zu öffnen), fährt mit SET fort (um den Variablenwert festzulegen) und endet mit PRINT (um das Etikett zu drucken). Weitere Informationen zu den verfügbaren Befehlen finden Sie im Abschnitt Benutzerdefinierte Befehle verwenden.

9.1.3.1. Beispiel für eine JOB-Befehlsdatei

Diese JOB-Datei öffnet das Etikett **`label2.nlbl`**, legt Variablenwerte fest und druckt ein einzelnes Etikett aus. Da kein PRINTER-Befehl verwendet wird, um den Druck umzuleiten, wird das Etikett auf dem im Etikett angegebenen Drucker gedruckt.

```
LABEL "label2.nlbl"
SET code="12345"
SET article="FUSILLI"
SET ean="383860026501"
SET weight="1,0 kg"
PRINT 1
```

9.1.4. XML-Befehlsdatei

Die in einer XML-Befehlsdatei verfügbaren Befehle sind eine Untermenge der NiceLabel Befehle. Die folgenden Befehle sind verfügbar: **LOGIN**, **LABEL**, **SET**, **PORT**, **PRINTER**, **SESSIONEND**, **SESSIONSTART** und **SESSIONPRINT**. Die Syntax für diese Befehle weicht in gewissem Maße ab, wenn sie in einer XML-Datei verwendet werden.

Das Stammelement in einer XML-Datei ist **<Nice_Commands>**. Das Element, das darauf folgen muss, ist **<Label>**. Dieses Element gibt an, welches Etikett verwendet werden soll.

Es gibt zwei Methoden, um den Etikettendruck zu starten: Regulärer Druck von Etiketten anhand des Elements **<Print_Job>** oder Druck von Etiketten im Rahmen einer Sitzung anhand des Elements **<Session_Print_Job>**. Sie können außerdem den Drucker ändern, auf dem die Etiketten gedruckt werden, und zusätzlich den Variablenwert festlegen.

9.1.4.1. Beispiel für eine XML-Befehlsdatei

Das folgende Beispiel dient als strukturelle Übersicht über die Elemente und ihre Attribute, die Sie in einer XML-Befehlsdatei verwenden können.

```
<nice_commands>
    <label name="label1.nlbl">

        <session_print_job printer="CAB A3 203DPI" skip=0 job_name="job name
1" print_to_file="filename 1">
            <session quantity="10">
                <variable name="variable name 1" >variable value 1</variable>
            </session>
        </session_print_job>

        <print_job printer="Zebra R-402" quantity="10" skip=0
identical_copies=1 number_of_sets=1 job_name="job name 2"
print_to_file="filename 2">
            <variable name="variable1" >1</variable>
            <variable name="variable2" >2</variable>
            <variable name="variable3" >3</variable>
```

```
        </print_job>
    </label>
</nice_commands>
```

Spezifikation für XML-Befehlsdateien

Dieser Abschnitt enthält eine Beschreibung der Struktur von XML-Befehlsdateien. Es gibt verschiedene Elemente, die Attribute enthalten. Einige Attribute sind erforderlich, andere sind optional. Einige Attribute können nur vordefinierte Werte annehmen, für andere können Sie benutzerdefinierte Werte festlegen.

- **<Nice_Commands>:** Dies ist ein Stammelement.
- **<Label>:** Gibt an, welche Etikettendatei geöffnet werden soll. Ist das Etikett bereits geöffnet, wird es nicht erneut geöffnet. Der Zugriff auf die Etikettendatei muss vom verwendeten Computer aus möglich sein. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk. Dieses Element kann innerhalb der Befehlsdatei mehrmals vorhanden sein.
    - **Name:** Dieses Attribut enthält den Etikettennamen. Sie können auch den Pfad in Kombination mit dem Namen angeben. Erforderlich.
- **<Print_Job>:** Dieses Element enthält Daten für einen einzelnen Etikettenauftrag. Dieses Element kann innerhalb der Befehlsdatei mehrmals vorhanden sein.
    - **Drucker:**: Verwenden Sie dieses Attribut, um den im Etikett definierten Drucker zu umgehen. Der Zugriff auf den Drucker muss vom verwendeten Computer aus möglich sein. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk. Optional.
    - **Quantity:** Verwenden Sie dieses Attribut, um die Anzahl zu druckender Etiketten festzulegen. Mögliche Werte: numerischer Wert, VARIABLE oder UNLIMITED. Weitere Informationen zu Parametern finden Sie im Abschnitt Etikett drucken. Erforderlich.
    - **Skip:** Verwenden Sie dieses Attribut, um die Anzahl von Etiketten festzulegen, die am Seitenanfang übersprungen werden sollen. Diese Funktion ist nützlich, wenn Sie Etiketten mit einem Laserdrucker auf einen Papierbogen drucken, der bereits teilweise bedruckt ist. Weitere Informationen finden Sie im Abschnitt Etikett drucken. Optional.
    - **Job_name:** Verwenden Sie dieses Attribut, um den Namen Ihrer Auftragsdatei festzulegen. Der angegebene Name wird im Druck-Spooler angezeigt. Weitere Informationen finden Sie im Abschnitt Druckauftragsnamen festlegen. Optional.
    - **Print_to_file:** Verwenden Sie dieses Attribut, um den Namen der Datei anzugeben, in der Sie die Druckerbefehle speichern möchten. Weitere Informationen finden Sie im Abschnitt Druck an Datei umleiten. Optional.
    - **Identical_copies:** Verwenden Sie dieses Attribut, um die Anzahl von Kopien festzulegen, die Sie für jedes Etikett benötigen. Weitere Informationen finden Sie im Abschnitt Etikett drucken. Optional.
- **<Session_Print_Job>:** Dieses Element enthält Befehle und Daten für eine oder mehrere Sitzungen. Das Element kann ein oder mehrere **<Session>**-Elemente enthalten. Es befolgt Regeln für den

Sitzungsdruck. Sie können dieses Element mehrmals innerhalb der Befehlsdatei verwenden. Für verfügbare Attribute, siehe Attribute für das **<Print_Job>**-Element. Alle davon sind gültig; Sie können lediglich das Quantity-Attribut nicht verwenden. In der Beschreibung für das Element **<Session>** erfahren Sie, wie Sie die Etikettenmenge für den Sitzungsdruck angeben können.

- **<Session>:** Dieses Element enthält Daten für eine Sitzung. Bei Verwendung des Sitzungsdrucks werden alle Etiketten zu einem einzelnen Druckauftrag codiert und an den Drucker gesendet.
    - **Quantity:** Verwenden Sie dieses Attribut, um die Anzahl zu druckender Etiketten festzulegen. Mögliche Werte: numerischer Wert, Zeichenfolge „VARIABLE“ oder Zeichenfolge „UNLIMITED“. Weitere Informationen zu Parametern finden Sie im Abschnitt Etikett drucken. Erforderlich.
- **<Variable>:** Dieses Element definiert die Variablenwerte auf dem Etikett. Dieses Element kann innerhalb der Befehlsdatei mehrmals vorhanden sein.
    - **Name:** Dieses Attribut enthält den Namen der Variablen. Erforderlich.

Definition des XML-Schemas (XSD) für XML-Befehlsdateien

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd"
elementFormDefault="qualified" xmlns="http://tempuri.org/XMLSchema.xsd"
xmlns:mstns="http://tempuri.org/XMLSchema.xsd" xmlns:xs="http://www.w3.org/
2001/XMLSchema">
    <xs:element name="nice_commands">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="label" maxOccurs="unbounded" minOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="print_job" maxOccurs="unbounded"
minOccurs="0">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="database"
maxOccurs="unbounded" minOccurs="0">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension
base="xs:string">
                                                        <xs:attribute
name="name" type="xs:string" use="required" />
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="table"
```

```
maxOccurs="unbounded" minOccurs="0">
                                        <xs:complexType>
                                            <xs:simpleContent>
                                                <xs:extension
base="xs:string">
                                                    <xs:attribute
name="name" type="xs:string" use="required" />
                                                </xs:extension>
                                            </xs:simpleContent>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element name="variable"
maxOccurs="unbounded" minOccurs="0">
                                        <xs:complexType>
                                            <xs:simpleContent>
                                                <xs:extension
base="xs:string">
                                                    <xs:attribute
name="name" type="xs:string" use="required" />
                                                </xs:extension>
                                            </xs:simpleContent>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                                <xs:attribute name="quantity"
type="xs:string" use="required" />
                                <xs:attribute name="printer"
type="xs:string" use="optional" />
                                <xs:attribute name="skip"
type="xs:integer" use="optional" />
                                <xs:attribute name="identical_copies"
type="xs:integer" use="optional" />
                                <xs:attribute name="number_of_sets"
type="xs:integer" use="optional" />
                                <xs:attribute name="job_name"
type="xs:string" use="optional" />
                                <xs:attribute name="print_to_file"
type="xs:string" use="optional" />
                                <xs:attribute name="print_to_file_append"
type="xs:boolean" use="optional" />
                                <xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="session_print_job"
maxOccurs="unbounded" minOccurs="0">
```

```
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="database"
maxOccurs="unbounded" minOccurs="0">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension
base="xs:string">
                                                        <xs:attribute
name="name" type="xs:string" use="required" />
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="table"
maxOccurs="unbounded" minOccurs="0">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension
base="xs:string">
                                                        <xs:attribute
name="name" type="xs:string" use="required" />
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="session"
minOccurs="1" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element
name="variable" minOccurs="0" maxOccurs="unbounded">
                                                        <xs:complexType>
                                                            <xs:simpleContent>
                                                                <xs:extension
base="xs:string">

<xs:attribute name="name" type="xs:string" use="required" /
>
                                                                </
xs:extension>
                                                            </xs:simpleContent>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                                <xs:attribute name="quantity"
```

```
type="xs:string" use="required" />
                                                </xs:complexType>
                                            </xs:element>
                                        </xs:sequence>
                                        <xs:attribute name="printer"
type="xs:string" use="optional" />
                                        <xs:attribute name="skip"
type="xs:integer" use="optional" />
                                        <xs:attribute name="job_name"
type="xs:string" use="optional" />
                                        <xs:attribute name="print_to_file"
type="xs:string" use="optional" />
                                        <xs:attribute name="print_to_file_append"
type="xs:boolean" use="optional" />
                                        <xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                            <xs:attribute name="name" type="xs:string"
use="required" />
                            <xs:attribute name="close" type="xs:boolean"
use="optional" />
                            <xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
                <xs:attribute name="quit" type="xs:boolean" use="required" />
            </xs:complexType>
        </xs:element>
    </xs:schema>
```

9.1.5. Oracle XML-Spezifikationen

Das Oracle XML-Format ist so ausgelegt, dass die XML-Inhalte interpretiert, geparst und auf Etiketten gedruckt werden können. Die XML-Dokumenttypdefinition (DTD) definiert die XML-Tags, die in einer XML-Datei verwendet werden sollen. Oracle generiert XML-Dateien gemäß dieser DTD, und Software anderer Anbieter übersetzt die XML-Datei ebenfalls gemäß der DTD.

Um eine Oracle XML-Befehlsdatei auszuführen, verwenden Sie die Aktion Oracle XML-Befehlsdatei ausführen.

9.1.5.1. XML-DTD

Nachfolgend finden Sie die XML-DTD, die verwendet wird, um eine XML sowohl für synchrone als auch für asynchrone XML-Formate zu bilden. Die DTD definiert Elemente, die in der XML-Datei verwendet werden, eine Liste ihrer Eigenschaften sowie Elemente der nächsten Ebene.

```
<!ELEMENT labels (label)*>
<!ATTLIST labels _FORMAT CDATA #IMPLIED>
<!ATTLIST labels _JOBNAME CDATA #IMPLIED>
<!ATTLIST labels _QUANTITY CDATA #IMPLIED>
<!ATTLIST labels _PRINTERNAME CDATA #IMPLIED>
<!ELEMENT label (variable)*>
<!ATTLIST label _FORMAT CDATA #IMPLIED>
<!ATTLIST label _JOBNAME CDATA #IMPLIED>
<!ATTLIST label _QUANTITY CDATA #IMPLIED>
```

9.1.5.2. Beispiel für eine Oracle XML-Datei

Diese Oracle XML-Datei stellt Daten für ein Etikett bereit (es gibt nur ein **<label>**-Element).

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE labels SYSTEM "label.dtd">
<labels _FORMAT ="Serial.nlbl" _QUANTITY="1" _PRINTERNAME="" _JOBNAME="Serial">
    <label>
        <variable name= "item">O Ring</variable>
        <variable name= "revision">V1</variable>
        <variable name= "lot">123</variable>
        <variable name= "serial_number">12345</variable>
        <variable name= "lot_status">123</variable>
        <variable name= "serial_number_status">Active</variable>
        <variable name= "organization">A1</variable>
    </label>
</labels>
```

Bei Ausführung dieser Oracle XML-Beispieldatei wird das Etikett **serial.nlbl** mit den folgenden Variablenwerten gedruckt.

Variablenname	Variablenwert
item	O Ring
revision	V1
lot	123
serial_number	12345

lot_status	123
serial_number_status	Active
organization	A1

Es wird **1** gedruckte Kopie des Etiketts mit dem Spooler-Auftragsnamen **Serial** geben. Der Druckername ist in der XML-Datei nicht angegeben; daher wird das Etikett auf dem Drucker gedruckt, der in der Etikettenvorlage definiert ist.

9.1.6. SAP AII XML-Spezifikationen

NiceLabel Automation kann als ein RFID-Gerätecontroller agieren, um RFID-Tags zu codieren und Etiketten zu drucken. Weitere Informationen zu SAP AII XML-Spezifikationen finden Sie im Dokument **SAP Auto-ID Infrastructure Device Controller Interface** auf der SAP-Website.

Um eine solche Befehlsdatei auszuführen, verwenden Sie die Aktion SAP AII XML-Befehlsdatei ausführen.

9.1.6.1. Beispiel für SAP AII XML

Diese SAP AII XML-Datei stellt Daten für ein Etikett bereit (beachten Sie, dass es nur ein **<label>**-Element gibt).

```
<?xml version="1.0" encoding="UTF-8"?>
<Command xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Command.xsd">
    <WriteTagData readerID="DEVICE ID">
        <Item>
            <FieldList format="c:\SAP Demo\SAP label.nlbl"
jobName="Writer_Device20040929165746" quantity="1">
            <Field name="EPC">00037000657330</Field>
            <Field name="EPC_TYPE">SGTIN-96</Field>
            <Field name="EPC_URN">urn:autoid:tag:sgtin:3.5.0037000.065774.8</
Field>
            <Field name="PRODUCT">Product</Field>
            <Field name="PRODUCT_DESCRIPTION">Product description</Field>
            </FieldList>
        </Item>
    </WriteTagData>
</Command>
```

Bei Ausführung dieser SAP AI XML-Beispieldatei wird das Etikett **c:\SAP Demo\SAP label.nlbl** mit den folgenden Variablenwerten gedruckt.

Variablenname	Variablenwert

EPC	00037000657330
EPC_TYPE	SGTIN-96
EPC	urn:autoid:tag:sgtin:3.5.0037000.065774.8
PRODUCT	Produkt
PRODUCT_DESCRIPTION	Produktbeschreibung

Es wird **1** gedruckte Kopie des Etiketts mit dem Spooler-Auftragsnamen **Writer_Device2004092916574** geben. Der Druckername ist in der XML-Datei nicht angegeben; daher wird das Etikett auf dem Drucker gedruckt, der in der Etikettenvorlage definiert ist.

9.2. Benutzerdefinierte Befehle

9.2.1. Benutzerdefinierte Befehle verwenden

NiceLabel-Befehle werden in Befehlsdateien verwendet, um den Etikettendruck zu steuern.NiceLabel Automation führt die Befehle in Befehlsdateien der Reihenfolge nach von oben nach unten aus. Weitere Details finden Sie im Abschnitt Spezifikationen für Befehlsdateien.

Sie können spezifische, benutzerdefinierte Befehle verwenden, wenn diese in Ihrem NiceLabel Automation-Produkt in Form von Aktionen zur Verfügung stehen.

Beispiel

Es ist möglich, den **SETPRINTPARAM**-Befehl zu verwenden, wenn Sie die Aktion **Druckparameter festlegen** auswählen können (in den Produktebenen Pro und Enterprise verfügbar).

Spezifikation der NiceLabel-Befehle

COMMENT

```
;
```

Beim Erstellen einer Befehlsdatei empfiehlt es sich, Ihre Befehle zu dokumentieren. Auf diese Weise können Sie leicht erkennen, welchen Zweck ein bestimmtes Skript erfüllt, wenn Sie sich den Code nach einiger Zeit erneut ansehen. Verwenden Sie zu Beginn der Zeile ein Semikolon (`;`). Alles, was darauf folgt, wird als Kommentar behandelt und nicht verarbeitet.

CLEARVARIABLEVALUES

```
CLEARVARIABLEVALUES
```

Dieser Befehl setzt Variablenwerte auf ihre Standardwerte zurück.

CREATEFILE

```
CREATEFILE <Dateiname> [, <Inhalt>]
```

Dieser Befehl erstellt eine Textdatei. Verwenden Sie sie, um einer Drittanwendung zu signalisieren, dass der Druckprozess begonnen hat oder abgeschlossen wurde, je nachdem, an welcher Stelle Sie den Befehl einsetzen. Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk.

DELETEFILE

```
DELETEFILE <Dateiname>
```

Löscht die angegebene Datei. Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk.

EXPORTLABEL

```
EXPORTLABEL ExportFileName [, ExportVariant]
```

Dieser Befehl automatisiert den Befehl „An Drucker exportieren“, der im Etikettendesigner zur Verfügung steht. Das Etikett wird direkt an den Drucker exportiert und zwecks Offline-Druck im Druckerspeicher abgelegt. Benutzer rufen das Etikett per Drucker-Tastenbefehl oder durch Senden einer Befehlsdatei an den Drucker auf. Dieselbe Funktionalität ist auch anhand der Aktion Etikett im Drucker speichern verfügbar.

ANMERKUNG

Um das zu exportierende Etikett anzugeben, verwenden Sie zuerst den Befehl **LABEL**.

- **ExportFileName:** Dieser obligatorische Parameter definiert den Dateinamen von erzeugten Druckerbefehlen.
- **ExportVariant:** Einige Drucker unterstützen mehrere Exportvarianten. Beim manuellen Export können Benutzer die gewünschte Exportvariante im Dialogfeld auswählen. Bei Nutzung des EXPORTLABEL-Befehls müssen Sie angeben, welche Exportvariante Sie verwenden möchten. Die Varianten sind im Etikettendesigner verfügbar, nachdem Sie den Speichern/Abrufen-Druckmodus aktiviert haben.
  Die erste Variante in der Liste hat den Wert 0. Die zweite Variante hat den Wert 1 usw.
  Falls Sie keine Variante angeben, wird standardmäßig der Wert 0 verwendet.

Weitere Informationen finden Sie im Abschnitt Speichern/Abrufen-Druckmodus verwenden im Benutzerhandbuch.

IGNOREERROR

```
IGNOREERROR <on> [,<off>]
```

Dieser Befehl gibt an, dass der Druckprozess beim Auftreten eines der folgenden Fehler in der JOB-Datei nicht beendet wird:

- Falscher Variablenname verwendet.
- Falscher Wert an Variable gesendet.
- Etikett ist nicht vorhanden / Zugriff nicht möglich.
- Drucker ist nicht vorhanden / Zugriff nicht möglich.

LABEL

```
LABEL <Etikettenname> [,<Druckername>]
```

Dieser Befehl öffnet das zu druckende Etikett. Wenn das Etikett bereits geladen ist, wird es nicht erneut geöffnet. Sie können auch den Pfad angeben. Schließen Sie den Etikettennamen in Anführungszeichen ein, falls er Leerzeichen enthält. Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk.

Der optionale Parameter **`Druckername`** gibt den Drucker an, für den das Etikett geöffnet wird. Verwenden Sie diese Einstellung, wenn Sie den Drucker umgehen wollen, der in der Etikettenvorlage gespeichert ist. Falls der Treiber für den angegebenen Druckernamen nicht installiert wurde oder nicht verfügbar ist, gibt der Befehl einen Fehler aus.

MESSAGEBOX

```
MESSAGEBOX <Meldung> [,<Titel>]
```

Dieser Befehl speichert eine benutzerdefinierte **Meldung** im Trigger-Protokoll. Falls die Meldung Leerzeichen oder Kommas enthält, müssen Sie den Text in Anführungszeichen (") einschließen.

Anschluss

```
PORT <Dateiname> [, APPEND]
```

Dieser Befehl übergeht die im Druckertreiber definierte Schnittstelle und leitet den Druck an eine Datei um. Falls der Pfad oder der Dateiname Leerzeichen enthält, schließen Sie den Wert in Anführungszeichen (") ein. Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk.

Der Parameter **APPEND** ist optional. Standardmäßig wird die Datei überschrieben. Verwenden Sie diesen Parameter, um die Daten an den Inhalt der vorhandenen Datei anzuhängen.

Wenn Sie den PORT-Befehl in der JOB-Datei verwenden, bleibt dieser bis zum nächsten PORT-Befehl oder bis zum Ende der Datei gültig (je nachdem, was vorher auftritt). Wenn Sie den PRINTER-Befehl nach Ausführung des PORT-Befehls verwenden, wird der für den ausgewählten Drucker festgelegte Port übergangen. Wenn Sie den tatsächlichen, für den ausgewählten Drucker definierten Port verwenden möchten, müssen Sie einen weiteren PORT-Befehl mit leerem Wert hinzufügen (**PORT = ""**).

PRINT

```
PRINT <Menge> [,<überspringen> [,<identische Etikettenkopien> [,Anzahl von
Etikettensätzen]]]
```

Dieser Befehl startet den Druckprozess.

- **Quantity:** Legt die Anzahl von gedruckten Etiketten fest.
  - **<number>:** Die angegebene Anzahl von Etiketten wird gedruckt.
  - **Variable einstellen:** Gibt an, dass eine Etikettenvariable als *variable Menge* definiert ist und die Anzahl zu druckender Etiketten enthält. Das Etikett bestimmt, wie viele Etiketten gedruckt werden.
  - **UNLIMITED:** Falls Sie eine Datenbank verwenden, um Werte für Objekte zu beziehen, werden bei unbegrenztem Druck so viele Etiketten gedruckt, wie Datensätze in der Datenbank vorhanden sind. Falls Sie keine Datenbank verwenden, wird die maximale vom Thermodrucker unterstützte Anzahl von Etiketten gedruckt.
- **Skip:** Gibt die Anzahl von Etiketten an, die auf der ersten Druckseite übersprungen werden sollen. Dieser Parameter wird beim Etikettendruck auf Papierbögen verwendet. Wenn ein Teil der Seite bereits bedruckt ist, können Sie denselben Bogen erneut verwenden, indem Sie die Startposition des ersten Etiketts verschieben.

- **Identische Etikettenkopien:** Gibt vor, wie viele Kopien desselben Etiketts gedruckt werden sollen.
- **Anzahl von Etikettensätzen.** Gibt an, wie viele Male der gesamte Druckprozess wiederholt werden soll.

ANMERKUNG

Geben Sie die Mengenwerte als numerischen Wert an, nicht als Zeichenfolgen-Wert. Stellen Sie den Wert nicht in Anführungszeichen.

PRINTER

```
PRINTER <Druckername>
```

Dieser Befehl übergeht den in der Etikettenvorlage definierten Drucker. Falls der Druckername Leerzeichen enthält, setzen Sie ihn in Anführungszeichen (**"**).

Verwenden Sie den Druckernamen, wie er in der Statuszeile der Etikettendesign-Anwendung angezeigt wird. Druckernamen entsprechen normalerweise den Namen unter „Drucker und Faxgeräte“ in der Systemsteuerung; dies ist aber nicht immer der Fall. Wenn Sie Netzwerkdrucker verwenden, wird der Name eventuell anhand der Syntax **`\\server\share`** angezeigt.

PRINTJOBNAME

```
PRINTJOBNAME
```

Dieser Befehl gibt den Namen des Druckauftrags an, der im Windows Spooler angezeigt wird. Falls der Name Leerzeichen oder Kommas enthält, müssen Sie den Wert in Anführungszeichen (**"**) einschließen.

SESSIONEND

```
SESSIONEND
```

Dieser Befehl schließt den Druckdatenstrom. Siehe auch **SESSIONSTART**.

ANMERKUNG

`SESSIONEND` Muss als einziges Element der Aktion „Benutzerdefinierte Befehle senden“ gesendet werden. Wenn Sie zusätzliche Befehle senden möchten, verwenden Sie separate „Benutzerdefinierte Befehle senden“-Aktionen.

SESSIONPRINT

```
SESSIONPRINT <Menge> [,<überspringen>]
```

Dieser Befehl druckt das aktuell referenzierte Etikett und fügt es dem momentan geöffneten Sitzungs-Druckdatenstrom hinzu. Sie können mehrere SESSIONPRINT-Befehle hintereinander verwenden und die referenzierten Etiketten in einem einzigen Druckdatenstrom vereinen. Der Datenstrom wird erst geschlossen, wenn Sie ihn anhand des SESSIONEND-Befehls schließen. Die Bedeutung der Parameter „Menge“ und „überspringen“ ist dieselbe wie im NiceCommand PRINT. Siehe auch den **SESSIONSTART**-Befehl.

- **Quantity:** Gibt die Anzahl von zu druckenden Etiketten an.
- **Skip:** Gibt die Anzahl von Etiketten an, die auf der ersten Druckseite übersprungen werden sollen. Dieser Parameter wird beim Etikettendruck auf Papierbögen verwendet. Wenn ein Teil der Seite bereits bedruckt ist, können Sie denselben Bogen erneut verwenden, indem Sie die Startposition des ersten Etiketts verschieben.

SESSIONSTART

```
SESSIONSTART
```

Dieser Befehl leitet den Sitzungsdruck ein.

Die drei Sitzungsdruck-bezogenen Befehle (**SESSIONSTART**, **SESSIONPRINT**, **SESSIONEND**) werden gemeinsam verwendet. Wenn Sie den Befehl PRINT verwenden, werden Daten für jedes Etikett in einem separaten Druckauftrag an den Drucker gesendet. Wenn Sie Etikettendaten für mehrere Etiketten in einen Druckdatenstrom integrieren möchten, sollten Sie die Befehle für Sitzungsdruck verwenden. Beginnen Sie zu diesem Zweck mit dem SESSIONSTART-Befehl, gefolgt von einer beliebigen Anzahl von SESSIONPRINT-Befehlen. Die Abfolge endet mit dem SESSIONEND-Befehl.

Nutzen Sie diese Befehle, um den Etikettendruckprozess zu optimieren. Das Drucken von Etiketten, die zu einem einzelnen Druckauftrag gehören, geht deutlich schneller als das Drucken von Etiketten anhand mehrerer Druckaufträge.

Verwenden Sie die folgenden Regeln, um sicherzustellen, dass der Sitzungsdruck nicht fehlschlägt:

- Sie können das Etikett innerhalb einer Sitzung nicht ändern.
- Sie können den Drucker innerhalb einer Sitzung nicht ändern.
- Sie müssen innerhalb einer Sitzung Werte für alle Variablen auf dem Etikett festlegen, selbst wenn einige der Variablen leere Werte haben.

SET

```
SET <Name>=<Wert>, [,<Schritt>[,<Anzahl der Wiederholungen>]]
```

Dieser Befehl weist der Variablen **`Name`** den **`Wert`** zu. Die Variable muss auf dem Etikett definiert sein; andernfalls wird ein Fehler ausgegeben. Befindet sich die Variable nicht auf dem Etikett, wird ein Fehler ausgegeben. **`Schritt`** und **`Anzahl der Wiederholungen`** sind Parameter für Zähler-Variablen. Diese

Parameter geben den Erhöhungswert für den Zähler sowie die Anzahl von gedruckten Etiketten vor Änderung des Zählerwerts an.

Falls ein Wert Leerzeichen oder Kommas enthält, müssen Sie den Text in Anführungszeichen (") setzen. Siehe auch **TEXTQUALIFIER**.

Falls Sie einen mehrzeiligen Wert zuweisen möchten, verwenden Sie **`\r\n`**, um einen Zeilenumbruch zu kodieren. **`\r`** wird durch CR (Schlittenrücklauf) und **`\n`** durch LF (Zeilenvorschub) ersetzt.

Seien Sie vorsichtig beim Festlegen von Variablen, welche Daten für Bilder auf dem Etikett angeben, da Backslash-Zeichen durch andere Zeichen ersetzt werden könnten.

Beispiel

Wenn Sie einer Variablen beispielsweise den Wert "c:\My Pictures\raw.jpg" zuweisen, wird "\r" durch das CR-Zeichen ersetzt.

SETPRINTPARAM

```
SETPRINTPARAM <paramname> = <Wert>
```

Mit diesem Befehl können Sie vor dem Drucken eine Feinabstimmung der Druckereinstellungen vornehmen. Die unterstützten Parameter für die Druckereinstellungen (**`paramname`**) sind:

- **PAPERBIN:** Gibt das Fach an, das die Etikettenmedien enthält. Falls der Drucker mit mehr als einem Papier-/Etikettenfach ausgestattet ist, können Sie festlegen, welches für den Druck verwendet werden soll. Der Name des Fachs sollte vom Druckertreiber bezogen werden.
- **PRINTSPEED:** Gibt die Druckgeschwindigkeit an. Die gültigen Werte variieren von Drucker zu Drucker. Im Druckerhandbuch finden Sie den exakten Wertebereich.
- **PRINTDARKNESS:** Legt die Drucktemperatur/den Druckkontrast fest. Die gültigen Werte variieren von Drucker zu Drucker. Im Druckerhandbuch finden Sie den exakten Wertebereich.
- **PRINTOFFSETX:** Legt den linken Versatz für alle Druckobjekte fest. Der Wert für diesen Parameter muss numerisch sein, positiv oder negativ, und die Anzahl von Punkten vorgeben.
- **PRINTOFFSETY:** Legt den oberen Versatz für alle Druckobjekte fest. Der Wert für diesen Parameter muss numerisch sein, positiv oder negativ, und die Anzahl von Punkten vorgeben.
- **PRINTERSETTINGS:** Gibt die benutzerdefinierten Druckereinstellungen an, die auf den Druckauftrag angewandt werden sollen. Der Parameter erfordert die gesamte DEVMODE-Struktur für den Zieldrucker; sie muss in Form einer Base64-codierten Zeichenfolge angegeben werden. Die DEVMODE enthält alle Parameter auf dem Druckertreiber (Geschwindigkeit, Temperatur, Versätze und andere). Weitere Informationen finden Sie im Abschnitt Informationen zu Druckereinstellungen und DEVMODE.

> **ANMERKUNG**
> Die Base64-codierte Zeichenfolge muss in Anführungszeichen (") stehen.

TEXTQUALIFIER

```
TEXTQUALIFIER <Zeichen>
```

Ein Textbegrenzer ist ein Zeichen, das einen Datenwert einschließt, welcher einer Variablen zugewiesen ist. Falls ein Datenwert Leerzeichen enthält, muss er in den Textbegrenzer eingeschlossen werden. Der Standard-Textbegrenzer ist das Anführungszeichen ("). Da das Anführungszeichen auch als Maßeinheit für Zoll verwendet wird, ist es in manchen Fällen schwierig, Daten mit Zollangaben in JOB-Dateien zu integrieren. Sie können zwei Anführungszeichen verwenden, um ein Anführungszeichen zu codieren, oder den Befehl TEXTQUALIFIER verwenden.

Beispiel

```
TEXTQUALIFIER %

SET Variable = %EPAK 12"X10 7/32"%
```

9.3. Zugriff auf freigegebene Ressourcen im Netzwerk

In diesem Abschnitt finden Sie die empfohlenen Schritte für die Nutzung von freigegebenen Ressourcen im Netzwerk.

- **Benutzerrechte für den Dienstmodus.** Die Ausführungskomponente von NiceLabel Automation läuft im Dienstmodus unter einem angegebenen Benutzerkonto und übernimmt die Zugriffsberechtigungen dieses Kontos. Damit NiceLabel Automation Etikettendateien öffnen und Druckertreiber nutzen kann, muss das verbundene Benutzerkonto dieselben Berechtigungen haben. Weitere Informationen finden Sie im Abschnitt Im Dienstmodus ausführen.
- **UNC-Notation für Netzwerkfreigaben.** Verwenden Sie beim Zugriff auf eine Datei auf einem Netzlaufwerk auf jeden Fall die UNC-Syntax (Universal Naming Convention) anstelle der zugeordneten Laufwerksbuchstaben. UNC ist eine Benennungskonvention, die Netzlaufwerke festlegt und zuordnet. NiceLabel Automation versucht, die Laufwerksbuchstaben-Syntax automatisch durch UNC-Syntax zu ersetzen.

Beispiel

Befindet sich eine Datei unter **`G:\Labels\label.nlbl`**, geben Sie sie in UNC-Notation als **`\\server\share\Labels\label.nlbl`** an (wobei das Laufwerk G: **`\\server\share`** zugeordnet ist).

- **Notation für den Zugriff auf Dateien in Control Center.** Wenn Sie eine Datei im Dokumentenspeicher in Control Center öffnen, können Sie HTTP-Notation wie **`http://servername:8080/label.nlbl`** oder WebDAV-Notation wie **`\\servername@8080\DavWWWRoot\label.nlbl`** nutzen.
  Weitere Hinweise:
  - Das Benutzerkonto, unter dem der NiceLabel Automation-Dienst ausgeführt wird, wird auch für den Abruf von Dateien aus dem Dokumentenspeicher verwendet. Dieser Benutzer muss in der Control Center Konfiguration eingerichtet sein. Dadurch wird sichergestellt, dass der Benutzer Zugriff auf die Dateien im Dokumentenspeicher hat.
  - WebDAV-Zugriff kann nur mit Windows-Benutzerauthentifizierung in Control Center genutzt werden.

  > **ANMERKUNG**
  > Der Dokumentenspeicher ist in den Produkten **LMS Enterprise** und **LMS Pro** verfügbar.

- **Verfügbarkeit von Druckertreibern.** Um Etiketten auf einem freigegebenen Netzwerkdrucker zu drucken, müssen Sie den Druckertreiber auf dem Server bereitstellen, auf dem NiceLabel Automation installiert ist. Stellen Sie sicher, dass das Benutzerkonto, unter dem der NiceLabel Automation Dienst ausgeführt wird, Zugriff auf den Druckertreiber hat. Wenn der Netzwerkdrucker gerade erst auf dem Rechner installiert wurde, ist er für NiceLabel Automation eventuell erst nach einem Neustart des Dienstes sichtbar. Um eine automatische Benachrichtigung über neue Netzwerkdruckertreiber zu ermöglichen, müssen Sie die entsprechende Regel für eingehenden Datenverkehr in der Windows Firewall aktivieren. Weitere Informationen finden Sie im Knowledge Base-Artikel.

9.4. Dokumentenspeicher und Versionierung von Konfigurationsdateien

Der Dokumentenspeicher ist eine Funktion von NiceLabel Control Center. Er ermöglicht es NiceLabel Control Center, als freigegebener Dateispeicher auf dem Server zu fungieren, in dem Benutzer ihre Dateien speichern, abrufen und ihre Versionen kontrollieren können.

Das Kontext-Tab **Dokumentenspeicher** ermöglicht Ihnen die Ausführung von Dokumentenspeicher-Aktionen direkt aus Automation Builder. So sind der Zugriff und das Öffnen der Automation Datei in NiceLabel Control Center unnötig.

> **ANMERKUNG**
> Dieses Kontext-Tab erfordert eine Verbindung mit NiceLabel Control Center. Für solche Konfigurationen ist eine LMS Enterprise-Lizenz erforderlich.

Die **Revisionierung**-Gruppe ermöglicht Ihnen die Ausführung der verfügbaren Dokumentenspeicher-Aktionen:

- **Auschecken:** Checkt die Datei auf dem NiceLabel Control Center Dokumentenspeicher aus und stellt sie zwecks Bearbeitung bereit. Die ausgecheckte Datei wird markiert und ihre Bearbeitungsmöglichkeiten werden für andere Benutzer gesperrt. Während der Autor an der Datei arbeitet, können alle anderen Benutzer nur die aktuelle Revision der Datei anzeigen.

ANMERKUNG

Nach Öffnen eines Dokuments aus dem Dokumentenspeicher (**Datei** > **Öffnen** > **Dokumentenspeicher**) kann es nicht bearbeitet werden, bevor Sie es auschecken.

- **Einchecken**: Checkt die Datei nach Abschluss der Bearbeitung im NiceLabel Control Center Dokumentenspeicher ein. Wenn Sie die Datei einchecken, wird die Revisionsnummer um eins erhöht. Der eingegebene Kommentar wird im Dateiprotokoll gespeichert.
- **Checkout verwerfen**: Ignoriert den Check-out der aktuellen Datei und anderen Benutzern vollständigen Zugriff auf sie.

WARNUNG

Wenn Sie auf **Checkout verwerfen** klicken, gehen alle Änderungen seit dem letzten Datei-Checkout verloren.

- **Dokumentenspeicher**: Öffnet den Speicherort des Dokumentenspeichers im verbundenen NiceLabel Control Center.

9.5. Zugriff auf Datenbanken

Wenn NiceLabel Automation Daten aus einer Datenbank abrufen muss, müssen Sie sicherstellen, dass im Windows-System der erforderliche Datenbanktreiber installiert ist. Datenbanktreiber werden von dem Unternehmen bereitgestellt, das die Datenbanksoftware entwickelt. Der Treiber, den Sie installieren, muss der Bit-Version Ihres Windows-Systems entsprechen. NiceLabel Software wird immer mit der Bitzahl Ihres Windows Systems ausgeführt.

9.5.1. 32-Bit-Windows

Wenn Sie eine 32-Bit-Version von Windows nutzen, können Sie nur 32-Bit-Datenbanktreiber installieren. Für die Konfiguration des Triggers in Automation Builder und für die Ausführung des Triggers im NiceLabel Automation-Dienst wird derselbe Datenbanktreiber verwendet. Alle NiceLabel Automation-Komponenten werden als 32-Bit-Anwendungen ausgeführt.

9.5.2. 64-Bit-Windows

Wenn Sie eine 64-Bit-Version von Windows nutzen, können Sie 32-Bit- und 64-Bit-Datenbanktreiber installieren. Die Anwendungen, die mit 64 Bit ausgeführt werden, nutzen 64-Bit-Datenbanktreiber. Die Anwendungen, die mit 32 Bit ausgeführt werden, nutzen 32-Bit-Datenbanktreiber.

Standardmäßig wird der Automation Dienst als 64-Bit-Prozess ausgeführt. In diesem Fall nutzt er 64-Bit-Datenbanktreiber, um eine Datenbankverbindung herzustellen. Wenn im System, in dem der Automation Dienst ausgeführt wird, keine 64-Bit-Datenbanktreiber verfügbar sind, wird die Datenbankverbindungs-Task an den **NiceLabel Proxy**-Prozess übergeben. Dieser Prozess wird immer als 32-Bit-Prozess ausgeführt.

9.6. Automatisches Ersetzen von Schriften

Sie können Ihre Etikettenvorlagen so erstellen, dass Textobjekte mit integrierten Druckerschriften gedruckt werden. Dies sind die Schriften, die im Speicher Ihres Druckers abgelegt sind. Wenn Sie versuchen, solche Etiketten auf einer anderen Art von Drucker zu drucken, sind die ausgewählten internen Schriften möglicherweise nicht verfügbar. Der andere Drucker unterstützt vermutlich eine völlig andere Reihe von internen Schriften. Diese Schriften können in einem solchen Fall ähnlich aussehen, aber unter anderen Namen verfügbar sein.

Eine fehlerhafte Schriftzuordnung kann auch eintreten, wenn die auf Ihren Etiketten verwendete Truetype-Schrift nicht auf dem Computer installiert ist, auf dem Sie Designer ausführen, um Etiketten zu erstellen und zu drucken.

Sie können Designer dafür konfigurieren, die Schriften auf Etiketten automatisch durch kompatible Schriften zu ersetzen. In solchen Fällen verwendet Designer die Namen der Schriften, um sie zuzuordnen und zu ersetzen. Wenn die Originalschrift nicht verfügbar ist, verwendet Designer die erste verfügbare Ersatzschrift, die in der Zuordnungstabelle definiert ist.

Wenn es keine geeigneten Ersatzschriften gibt, nutzt Designer die Schrift Arial Truetype.

ANMERKUNG

Nach Konfiguration der Schriftersetzung werden die Zuordnungsregeln ausgeführt, wenn Sie den Drucker für Ihr Etikett ändern.

Schriftzuordnung konfigurieren

1. Öffnen Sie den Dateiexplorer und navigieren Sie zum folgenden Ordner:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10
```

2. Kopieren Sie die Datei **fontmapping.def** in **fontmapping.local.def**.
3. Öffnen Sie die Datei **fontmapping.local.def** im XML-Editor Ihrer Wahl.
4. Erstellen Sie im Element **FontMappings** ein neues Element mit benutzerdefiniertem Namen.
5. Erstellen Sie innerhalb des neuen Elements mindestens zwei Elemente mit dem Namen **Mapping**.
    - Der Wert des ersten Elements namens Mapping muss den Namen der Originalschrift enthalten.
    - Der Wert des zweiten Elements namens Mapping muss den Namen der Ersatzschrift enthalten.

ANMERKUNG

Weitere Mapping-Elemente mit neuen Schriftnamen sind nicht erlaubt. Ist die erste Ersatzschrift nicht verfügbar, versucht Designer, die nächste zu laden. Ist keine Ersatzschrift verfügbar, wird stattdessen Arial TrueType verwendet.

ANMERKUNG

Die Datei **fontmapping.local.def** gehört Ihnen und bleibt bei den Upgrades erhalten. Die Datei **fontmapping.def** gehört jedoch zu NiceLabel und wird bei den Upgrades überschrieben. Nehmen Sie keine Änderungen an der Datei **fontmapping.def** vor.

Beispiel für eine Mapping-Konfiguration

Im folgenden Beispiel werden zwei Schrift-Zuordnungen konfiguriert.

- Die erste Zuordnung konvertiert **Avery**-Schriften in eine passende **Novexx**-Schrift. Die Schrift **Avery YT100** wird zum Beispiel durch **Novexx YT100** ersetzt, und die Schrift **Avery 1** durch **Novexx 1**. Falls keine Novexx-Schrift verfügbar ist, wird **Arial** TrueType verwendet.
- Die zweite Zuordnung konvertiert die Schrift **Avery YT100** in **Novexx YT104**. Ist diese Schrift nicht verfügbar, wird die Schrift **Zebra 0** verwendet. Wenn diese Schrift ebenfalls nicht verfügbar ist, , wird **Arial** TrueType verwendet.
- Die zweite Zuordnung ist der ersten übergeordnet.

```
<?xml version="1.0" encoding="utf-8"?>
<FontMappings>
    <AveryNovexx>
        <Mapping>Avery</Mapping>
        <Mapping>Novexx</Mapping>
    </AveryNovexx>
    <TextReplacement>
        <Mapping>Avery YT100</Mapping>
        <Mapping>Novexx YT104</Mapping>
        <Mapping>Zebra 0</Mapping>
    </TextReplacement>
</FontMappings>
```

9.7. Berichte automatisieren

PRODUKTEBENEN-INFO

Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Kombinieren Sie Ihre Geschäftssystemdaten mit Etiketten, die **Berichte** enthalten. Erstellen Sie Berichte in NiceLabel Designer und füllen Sie Ihre Berichte mit Daten aus Ihren Geschäftssystemen in NiceLabel Automation.

In Ihren ausgefüllten Berichten werden Ihre Geschäftssystemdaten zum Drucken verwendet. NiceLabel Automation:

- **Empfängt** Ihre Geschäftssystem**-Daten**.
- **Zerlegt Ihre Daten** mit einem **Datenfilter**.

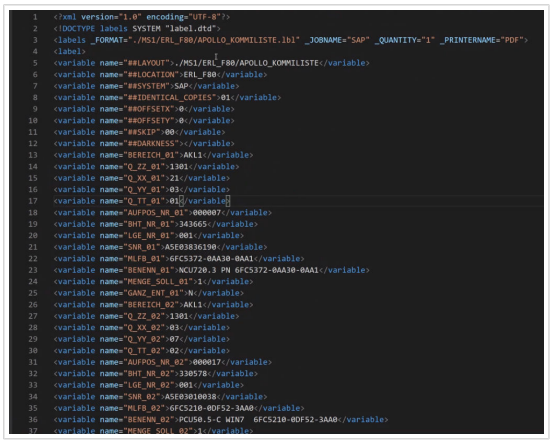
- **Füllt Ihren Bericht** mit zerlegten Daten.
- **Führt den Druck** für Ihre neu ausgefüllten Berichte mit einem **Trigger** aus.

9.7.1. Temporäre Datenbanken erstellen

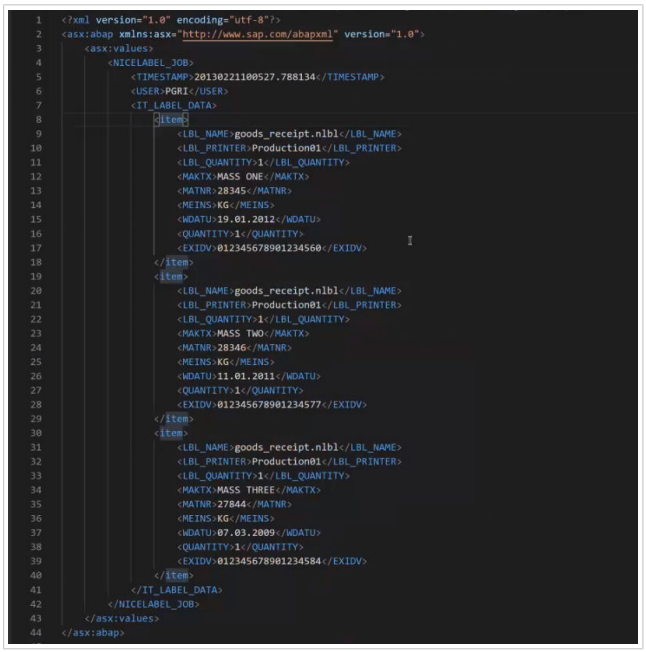
Sie müssen Ihre **Berichte** entwerfen, bevor Sie sie automatisieren. Um Berichte zu entwerfen, müssen Sie sie mit einer **Datenbank** verknüpfen.

Wenn Sie Daten aus externen Geschäftssystemen für Berichte verwenden, haben Sie **keine Datenbank**, mit der Sie arbeiten können. Für eine korrekte Funktionsweise benötigen Berichtsdaten eine **hierarchische Struktur** mit klar definierten Elementen.

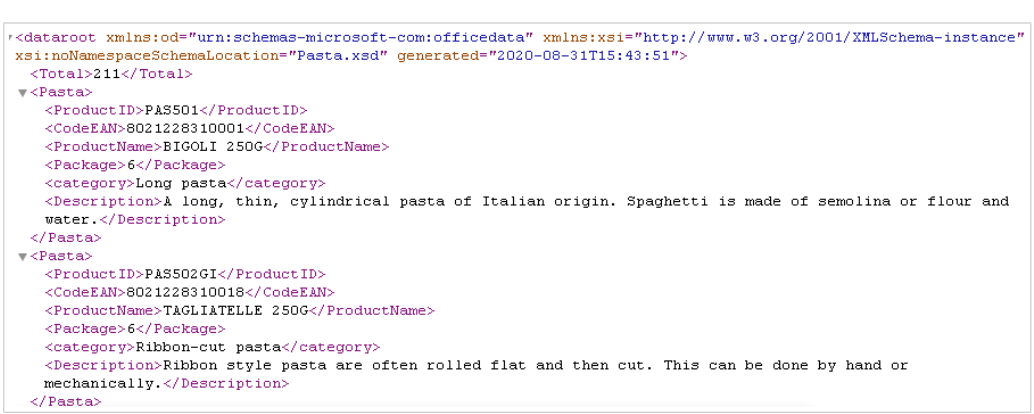
Erstellen Sie eine **temporäre Datenbank** mit Daten aus Ihrem Geschäftssystem (normalerweise XML- oder JSON-Daten). Entwerfen und konfigurieren Sie Ihre Berichte mit Ihrer temporären Datenbank. **Ihre temporäre Datenbank dient nur Einrichtungszwecken**. Konfigurierte Berichte werden mit Automation-Triggern, nicht aber mit Ihrer temporären Datenbank gedruckt.



Daten aus Ihrem Geschäftssystem ohne Hierarchie.



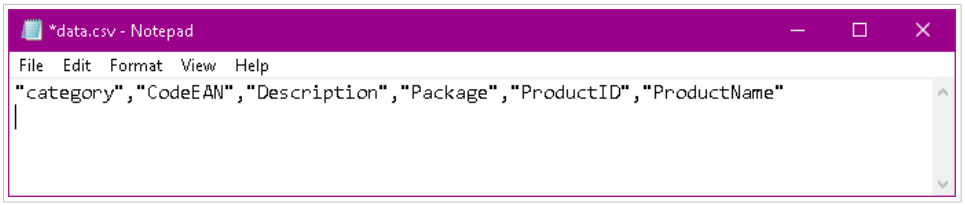
Daten mit Hierarchie, mit denen Sie Berichte drucken können.



Beispiel für XML-Daten, die Sie zum Entwerfen von Berichten benötigen (nicht analysiert).

Erstellen Sie eine CSV-Textdatenbankdatei für Ihre temporäre Datenbank. Sie haben mehrere Optionen:

1. Konvertieren Sie Ihre Datenstruktur manuell in eine CSV-Textdatei:



Temporäre CSV-Textdatenbanken manuell erstellen.

2. Verwenden Sie ein gängiges Konvertierungstool für die Konvertierung von XML in Excel oder XML in CSV und formatieren Sie Ihre Daten manuell in einer Tabelle.

3. Verwenden Sie einen Automation-**Datenfilter**, um Ihre Daten automatisch in eine CSV-Textdatendatei zu zerlegen (Weitere Informationen finden Sie in Ihrer beigefügten **XML zu CSV**-Musterdatei).

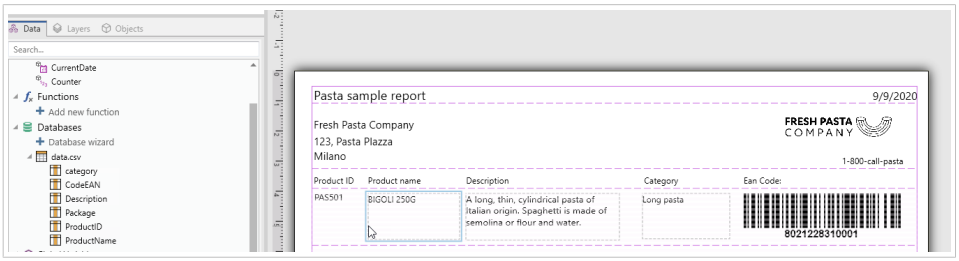
9.7.2. Automatisierte Berichte entwerfen

Öffnen Sie den **Designer**, um Ihren **Bericht** mit Ihrer temporären CSV-Textdatenbank zu erstellen:

1. **Verknüpfen Sie** Ihre temporäre CSV-Textdatenbank mit Ihrem**Bericht**.

2. **Entwerfen Sie** Ihren Bericht mithilfe Ihrer CSV-Textdaten als Variablen für Objekte in Ihrer **Repeater-Definition**.

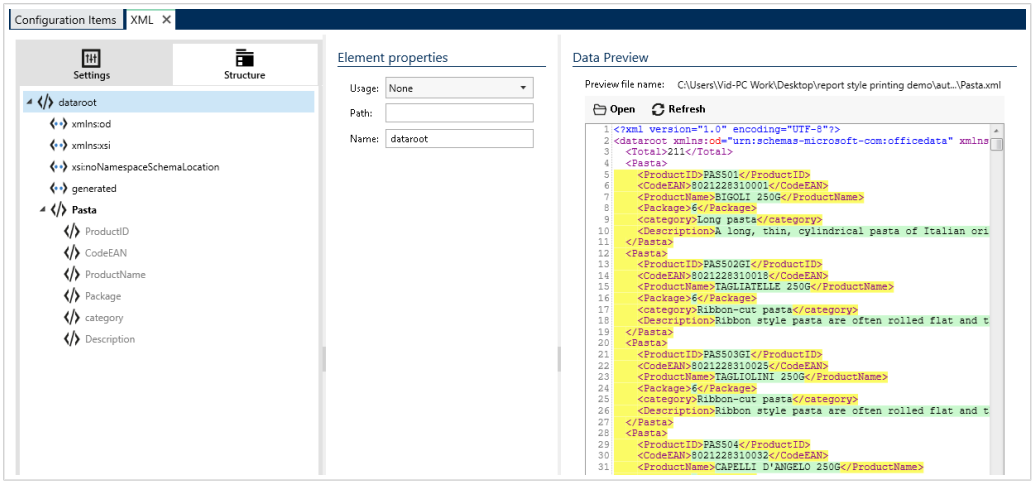
> **ANMERKUNG**
>
> **Gleichen Sie** Ihre **Variablennamen** mit den **Datennamen** in Ihrem Datenfilter ab, um Ihre Daten korrekt abzubilden.



Berichte mit Ihrer temporären CSV-Textdatenbank entwerfen.

9.7.3. Datenfilter erstellen

Erstellen Sie in Automation einen **Datenfilter**, der in Ihrem Bericht verwendet werden soll (weitere Informationen zum Erstellen von XML-Filtern finden Sie in Ihrem Beispielbeispiel):



Ihren XML-Datenfilter konfigurieren.

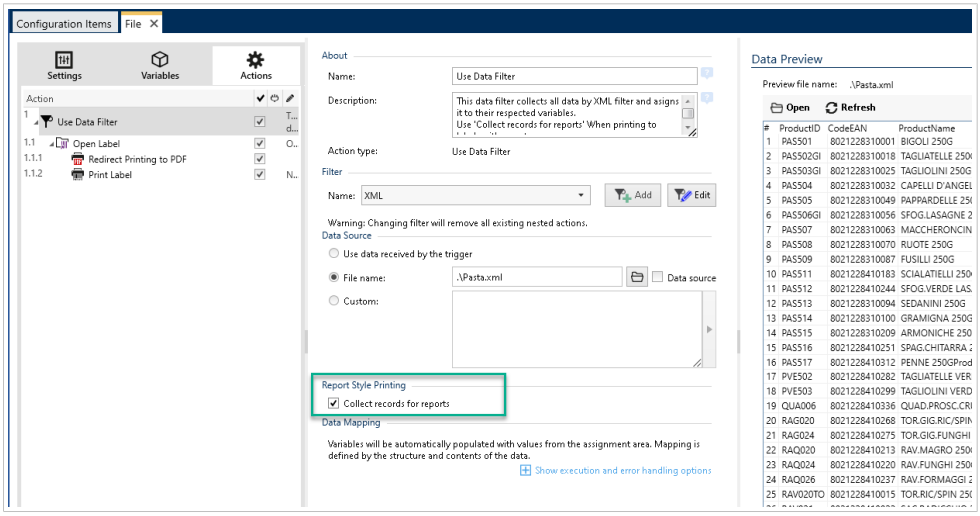
Tipps zum Erstellen von Datenfiltern für Berichte:

- Sie können den **Strukturimport-Assistent** zum Importieren Ihrer Datenstruktur verwenden.
- Benennen Sie Ihre Elemente in Datenblöcken wiederholbarer Elemente.
- Legen Sie Ihre Elemente als Zuordnungsbereiche oder Unterelemente als Variablen fest.

9.7.4. Trigger für Ihren neuen Datenfilter erstellen

> **ANMERKUNG**
> Damit Ihr Trigger funktioniert, müssen Ihre Datenquellennamen in Designer mit den Datenblockfeldnamen in Ihrem Datenfilter übereinstimmen.

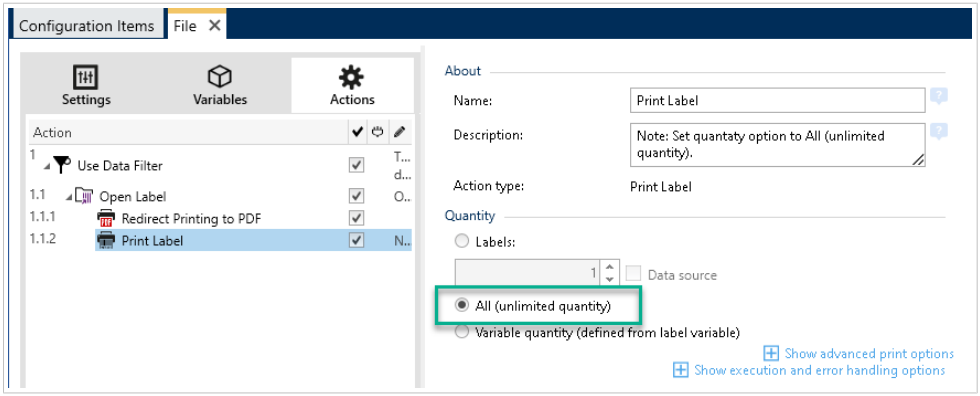
- Führen Sie den Vorgang **Datenfilter verwenden** aus, um Ihren Datenfilter anzuwenden.



- Aktivieren Sie **Datensätze für Bericht erfassen**, um alle Datensätze in einer Tabelle zu erfassen, die von der Automation zum Öffnen Ihrer Berichtetiketten verwendet.

> **ANMERKUNG**
> Verwenden Sie in Ihren Berichten, Triggern und Filtern dieselben Datenstrukturnamen.

- Führen Sie den Vorgang **Etikett drucken** aus.



  - Aktivieren Sie **Alle (unbegrenzte Menge)**.

Führen Sie Ihren Trigger aus. Ihr Bericht wird jetzt automatisch mit neuen Daten aktualisiert und mit Ihrem Trigger gedruckt.

9.8. Standardeinstellungen für Multi-Thread-Druck ändern

> **PRODUKTEBENEN-INFO**
> Diese Funktion ist verfügbar in **LMS Enterprise** und **LMS Pro** .

Jedes NiceLabel Automation-Produkt kann mehrere Prozessorkerne nutzen – jeder Kern führt einen unabhängigen Druckprozess aus. Die Hälfte der Kerne wird zur Verarbeitung gleichzeitiger *normaler* Threads, die andere Hälfte zur Verarbeitung gleichzeitiger *Sitzungsdruck*-Threads verwendet.

> **ANMERKUNG**
> Unter normalen Umständen müssen Sie die Standardeinstellungen nicht ändern. Ändern Sie diese Standardeinstellungen nur, wenn Sie sich mit der Materie auskennen.

So ändern Sie die Anzahl gleichzeitiger Druck-Threads:

1. Öffnen Sie die Datei **`product.config`** in einem Text-Editor.
   Die Datei befindet sich hier:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
```

2. Ändern Sie die Werte für die Elemente **MaxConcurrentPrintProcesses** und **MaxConcurrentSessionPrintProcesses**.

```
<configuration>
    <IntegrationService>
        <MaxConcurrentPrintProcesses>1</MaxConcurrentPrintProcesses>
        <MaxConcurrentSessionPrintProcesses>1</
MaxConcurrentSessionPrintProcesses>
    </IntegrationService>
</configuration>
```

3. Speichern Sie die Datei. NiceLabel Automation aktualisiert den Dienst automatisch mit der neuen Anzahl von Druck-Threads.

9.9. Kompatibilität mit NiceWatch-Produkten

NiceLabel Automation Ermöglicht es Ihnen, Trigger-Konfigurationen zu laden, die mit älteren NiceWatch Produkten erstellt wurden. In den meisten Fällen können Sie die NiceWatch-Konfiguration anhand von NiceLabel Automation ausführen, ohne Anpassungen vorzunehmen.

NiceLabel Automation-Produkte nutzen eine neue .NET-basierte Druck-Engine, die für Performance und geringe Arbeitsspeicherbelastung optimiert wurde. Die neue Druck-Engine unterstützt nicht alle Etikettendesign-Optionen, die im Etiketten-Designer zur Verfügung stehen. Diese Lücken werden in jeder neuen Version von NiceLabel Automation nach und nach gefüllt, aber es gibt möglicherweise immer noch einige nicht verfügbare Funktionen.

Kompatibilitätsprobleme beheben

NiceLabel Automation warnt Sie, wenn Sie versuchen, vorhandene Etikettenvorlagen zu drucken, die Funktionen beinhalten, welche von der neuen Druck-Engine nicht unterstützt werden.

Automation benachrichtigt Sie über folgende Inkompatibilitäten mit der NiceWatch-Konfiguration oder mit Etikettenvorlagen:

- **Kompatibilität mit der Trigger-Konfiguration:** Wenn Sie die NiceWatch-Konfiguration (.MIS file) öffnen, gleicht NiceLabel Automation sie mit den unterstützten Funktionen ab. Nicht alle Funktionen von NiceWatch stehen in NiceLabel Automation zur Verfügung. Einige sind überhaupt nicht verfügbar, während andere nur unterschiedlich konfiguriert werden. Wenn die MIS-Daten nicht unterstützte Funktionen enthält, werden diese aufgelistet. Automation entfernt diese Funktionen aus der Konfiguration.
  In solchen Fällen müssen Sie die .MIS-Datei in Automation Builder öffnen und die Kompatibilitätsprobleme beheben. Sie müssen die verfügbaren Funktionen von NiceLabel Automation nutzen, um die entfernten Konfigurationselemente neu zu erstellen.
- **Kompatibilität mit Etikettenvorlagen:** Wenn Ihre vorhandenen Etikettenvorlagen Druck-Engine-Funktionen enthalten, die von NiceLabel Automation nicht unterstützt werden, werden Fehlermeldungen im **Log**-Bereich angezeigt. Diese Information wird im Automation Builder (beim Erstellen von Triggern) oder in Automation Manager (beim Ausführen von Triggern) angezeigt.
  In diesem Fall müssen Sie die Etikettendatei im Etiketten-Designer öffnen und die nicht unterstützten Funktionen aus dem Etikett entfernen.

ANMERKUNG

Weitere Informationen über Kompatibilitätsprobleme mit NiceWatch und Etiketten-Designern finden Sie im NiceLabel Automation Compatibility with NiceLabel Designers V6 and NiceWatch V5.

NiceWatch-Konfiguration zur Bearbeitung öffnen

Öffnen Sie die vorhandene NiceWatch-Konfiguration (.MIS Datei) in Automation Builder und bearbeiten Sie sie in Automation Builder. Sie können die Konfiguration nur als MISX-Datei speichern.

So bearbeiten Sie die NiceWatch-Konfiguration:

1. Starten Sie Automation Builder.
2. Wählen Sie **Datei** > **NiceWatch Datei öffnen**.
3. Wählen Sie im Dialogfeld **Öffnen** die gewünschte NiceWatch-Konfigurationsdatei (.MIS Datei) aus.
4. Klicken Sie auf **OK**.
5. Wenn die Konfiguration nicht unterstützte Funktionen beinhaltet, werden sie in einer Liste angezeigt. Automation entfernt sie aus der Konfiguration.

NiceWatch-Konfiguration zur Ausführung öffnen

Sie können die NiceWatch-Konfiguration (.MIS-Datei) in Automation Manager öffnen, ohne sie ins NiceLabel Automation Dateiformat (.MISX-Datei) zu konvertieren. Wenn die Trigger aus NiceWatch mit NiceLabel Automation kompatibel sind, können Sie sie auf Anhieb verwenden.

Um die NiceWatch-Konfiguration zu öffnen und zu implementieren, tun Sie Folgendes:

1. Starten Sie Automation Manager.
2. Klicken Sie auf die **Hinzufügen (+)**-Schaltfläche.
3. Ändern Sie im Dialogfeld **Öffnen** den Dateityp zu **NiceWatch Konfiguration**.
4. Navigieren Sie zur NiceWatch-Konfigurationsdatei (.MIS-Datei).
5. Klicken Sie auf **OK**.
6. Automation Manager zeigt den Trigger für die ausgewählte Konfiguration an. Um den Trigger zu starten, wählen Sie ihn aus und klicken Sie auf **Start**.

ANMERKUNG

Falls Kompatibilitätsprobleme mit der NiceWatch-Konfiguration bestehen, müssen Sie sie in Automation Builder öffnen und umkonfigurieren.

9.10. Automation-Dienst mit Befehlszeilenparametern steuern

In diesem Abschnitt erfahren Sie, wie Sie die Eingabeaufforderung für folgenden Zwecke nutzen:

- Automation-Dienste starten oder anhalten.
- Steuern, welche Konfigurationen geladen werden.
- Steuern, welche Trigger aktiv sind.

ANMERKUNG

Stellen Sie sicher, dass Sie die **Eingabeaufforderung** mit erhöhten Rechten (Administratorrechten) ausführen. Klicken Sie mit der rechten Maustaste auf und wählen Sie Als **Administrator ausführen**.

Dienste starten und anhalten

Um beide Dienste zu starten, verwenden Sie die folgenden Befehle:

```
net start NiceLabelProxyService2019

net start NiceLabelAutomationService2019
```

Wenn Sie beim Starten des Dienstes die Konfigurationsdatei öffnen möchten, verwenden Sie:

```
net start NiceLabelAutomationService2019 [Konfiguration]
```

Zum Beispiel:

```
net start NiceLabelAutomationService2019 "c:\Project\configuration.MISX"
```

Um Dienste anzuhalten, verwenden Sie die folgenden Befehle:

```
net stop NiceLabelProxyService2019

net stop NiceLabelAutomationService2019
```

Konfigurationen und Trigger verwalten

Der NiceLabel Automation-Dienst kann mithilfe der Automation Manager Befehlszeilenparameter gesteuert werden. Verwenden Sie die folgende allgemeine Syntax:

```
NiceLabelAutomationManager.exe COMMAND Konfiguration [TriggerName] [/SHOWUI]
```

ANMERKUNG

Schließen Sie den vollständigen Pfad zum Konfigurationsnamen ein. Verwenden Sie nicht nur den Dateinamen.

Konfiguration hinzufügen

Die angegebene Konfiguration wird im Dienst geladen. Kein Trigger wird gestartet. Wenn Sie den Parameter **`/SHOWUI`** einschließen, wird die Automation Manager-Benutzeroberfläche gestartet.

```
NiceLabelAutomationManager.exe ADD c:\Project\configuration.MISX /SHOWUI
```

Konfiguration neu laden

Die angegebene Konfiguration wird erneut im Dienst geladen. Der Ausführungsstatus aller Trigger wird beibehalten. Ein erneutes Laden der Konfiguration erzwingt eine Aktualisierung aller für diese Konfiguration zwischengespeicherten Dateien. Weitere Informationen finden Sie in den Abschnitten Dateien zwischenspeichern. Wenn Sie den Parameter **`/SHOWUI`** einschließen, wird die Automation Manager-Benutzeroberfläche gestartet.

```
NiceLabelAutomationManager.exe RELOAD c:\Project\configuration.MISX /SHOWUI
```

Konfiguration entfernen

Die angegebene Konfiguration und alle ihre Trigger werden aus dem Dienst entfernt.

```
NiceLabelAutomationManager.exe REMOVE c:\Project\configuration.MISX
```

Einen Trigger starten

Der angegebene Trigger wird in der bereits geladenen Konfiguration gestartet.

```
NiceLabelAutomationManager.exe START c:\Project\Configuration.MISX CSVTrigger
```

Einen Trigger anhalten

Der angegebene Trigger wird in der bereits geladenen Konfiguration angehalten.

```
NiceLabelAutomationManager.exe STOP c:\Project\configuration.MISX CSVTrigger
```

Status-Codes
Status-Codes bieten Feedback zur Befehlszeilenausführung. Um die Ausgabe von Status-Codes zu aktivieren, verwenden Sie die folgende Befehlszeilensyntax:

```
start /wait NiceLabelAutomationManager.exe COMMAND Konfiguration [TriggerName]
[/SHOWUI]
```

Die Status-Codes werden in der **`errorlevel`**-Systemvariablen erfasst. Um den Status-Code anzuzeigen, führen Sie den folgenden Befehl aus:

```
echo %errorlevel%
```

Liste von Status-Codes:

Status-Code	Beschreibung
0	Kein Fehler aufgetreten
100	Konfigurationsdatei nicht gefunden
101	Konfiguration kann nicht geladen werden
200	Trigger nicht gefunden
201	Trigger kann nicht gestartet werden

Benutzer-Anmeldedaten zur Authentifizierung bei der Anwendung bereitstellen
Wenn Sie das LMS Enterprise- oder LMS Pro-System für die Verwendung der **Anwendungs-Authentifizierung** (nicht zur **Windows-Authentifizierung**) konfiguriert haben, müssen Sie den Benutzerkonten ausreichend Berechtigungen zur Verwaltung der Konfigurationen und Trigger zuweisen.

Sie können hierfür zwei Befehlszeilenparameter verwenden:

- -USER:**`[Benutzername]`**. Dabei ist **`[Benutzername]`** ein Platzhalter für den tatsächlichen Benutzernamen.
- -PASSWORD:**`[Passwort]`**. Dabei ist **`[Passwort]`** ein Platzhalter für das tatsächliche Passwort.

9.11. Ersetzen der Datenbank-Verbindungszeile

Eine Konfigurationsdatei für den Automation-Dienst kann Befehle zum Ersetzen der Datenbank-Verbindungszeile enthalten.

Sie können den Dienst so konfigurieren, dass bestimmte Teile der Verbindungszeile während der Ausführung des Triggers ersetzt werden. Dies befähigt eine einzelne Instanz von Automation, dieselbe Konfiguration auszuführen, aber einen unterschiedlichen Datenbankserver für datenbankbasierte Funktionen zu nutzen. So können Benutzer Trigger in Entwicklungsumgebungen konfigurieren und ohne Änderungen an der Konfiguration in der Produktionsumgebung ausführen.

Die Logik zur Ersetzung der Verbindungszeile wird in der Datei **`DatabaseConnections.Config`** festgelegt, die sich im Automation-Systemordner befindet.

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10
```

Die Konfigurationsdatei definiert From-To-Paare (alt-neu) anhand ihrer XML-Struktur. Das **`<Replacement>`**-Element enthält ein **`<From>`**- und ein **`<To>`**-Element. Während der Triggerausführung wird die „From“-Zeichenfolge durch die „To“-Zeichenfolge ersetzt. Sie können beliebig viele **`<Replacement>`**-Elemente definieren.

Die Konfigurationsdatei wird nicht zusammen mit Automation installiert. Sie können sie selbst anhand einer Struktur hinzufügen, die im folgenden Beispiel gezeigt wird. Dieselben Regeln für Suchen und Ersetzen werden auf alle Trigger angewandt, die im Automation-Dienst auf diesem Rechner ausgeführt werden.

ANMERKUNG
Starten Sie beide Automation-Dienste neu, nachdem Sie die config-Datei zum Automation-Systemordner hinzugefügt haben.

Beispiel

Ein vorhandener Trigger beinhaltet eine Verbindung zu einem Microsoft SQL-Server namens **`mySQLServer`** und einer Datenbank namens **`myDatabase`**. Sie wollen die Verbindungszeile so aktualisieren, dass die Datenbank **`NEW_myDatabase`** auf dem Server **`NEW_mySQLServer`** verwendet wird.

Es müssen zwei Ersetzungselemente definiert werden– eines zur Änderung des Servernamens und eines zur Änderung des Datenbanknamens.

```
<?xml version="1.0" encoding="UTF-8"?>
<DatabaseConnectionReplacements>
    <Replacement>
        <From>Data Source=mySQLServer</From>
        <To>Data Source=NEW_mySQLServer</To>
    </Replacement>
    <Replacement>
        <From>Initial Catalog=myDatabase</From>
        <To>Initial Catalog=NEW_myDatabase</To>
    </Replacement>
</DatabaseConnectionReplacements>
```

9.12. Eingabe von Sonderzeichen (Steuercodes)

Sonderzeichen oder Steuercodes sind Binärzeichen, die nicht auf der Tastatur vertreten sind. Sie können sie nicht wie normale Zeichen eingeben, sondern müssen sie mit einer speziellen Syntax codieren. Sie müssen solche Zeichen verwenden, wenn Sie mit Geräten mit serieller Schnittstelle kommunizieren, Daten am TCP/IP-Port empfangen oder mit Binärdateien (z. B. Druckdateien) arbeiten.

Es gibt zwei Methoden zur Eingabe von Sonderzeichen:

- **Geben Sie die Zeichen manuell ein**, indem Sie eines der beschriebenen Syntax-Beispiele nutzen:
    - Verwenden Sie die Syntax **`<special_character_acronim>`**, z. B. **`<FF>`** für Seitenvorschub oder **`<CR>`** für Schlittenrücklauf oder **`<CR><LF>`** für Zeilenumbruch.
    - Verwenden Sie die Syntax **`<#hex_code>`**, z. B. **`<#0D>`** (13 in Dezimalschreibweise) für Schlittenrücklauf oder **`<#00>`** für Nullzeichen.

  Weitere Informationen finden Sie in den Abschnitten Liste mit Steuercodes.
- **Fügen Sie die Zeichen aus der Liste ein.** Objekte, die Sonderzeichen als Inhalte unterstützen, werden mit einer Pfeilschaltfläche auf ihrer rechten Seite angezeigt. Die Schaltfläche öffnet eine Liste mit allen verfügbaren Sonderzeichen. Wenn Sie ein Zeichen in der Liste auswählen, wird es zum Inhalt hinzugefügt. Weitere Informationen finden Sie im Abschnitt Zusammengesetzte Werte verwenden im Benutzerhandbuch.

9.13. Liste mit Steuercodes

ASCII-Code	Abkürzung	Beschreibung
1	SOH	Beginn der Kopfzeile
2	STX	Beginn der Nachricht
3	ETX	Ende der Nachricht
4	EOT	Ende der Übertragung
5	ENQ	Anfrage
6	ACK	Positive Bestätigung
7	BEL	Tonzeichen
8	BS	Rückschritt
9	HT	Horizontaler Tabulator
10	LF	Zeilenvorschub
11	VT	Vertikaler Tabulator
12	FF	Seitenvorschub
13	CR	Wagenrücklauf
14	SO	Umschaltung
15	SI	Rückschaltung
16	DLE	Datenverbindungs-Fluchtsymbol
17	DC1	XON – Gerätekontrollzeichen 1
18	DC2	Gerätekontrollzeichen 2
19	DC3	XOFF – Gerätekontrollzeichen 3
20	DC4	Gerätekontrollzeichen 4
21	NAK	Negative Bestätigung
22	SYN	Synchronisierungssignal
23	ETB	Ende des Übertragungsblocks
24	CAN	Abbruch
25	EM	Ende des Mediums
26	SUB	Ersatz
27	ESC	Fluchtsymbol
28	FS	Dateitrenner
29	GS	Gruppentrenner
30	RS	Datensatztrenner
31	US	Einheitentrenner
188	FNC1	Funktionscode 1
189	FNC2	Funktionscode 2

190	FNC3	Funktionscode 3
191	FNC4	Funktionscode 4

9.14. Lizenzierung und Druckernutzung

Je nach Lizenztyp ist Ihr NiceLabel Produkt eventuell auf eine bestimmte Anzahl von Druckern beschränkt, die gleichzeitig verwendet werden können. Im Fall einer Mehrbenutzer-Lizenz speichert NiceLabel die Anzahl und Namen der verschiedenen Drucker, die Sie zum Drucken auf allen NiceLabel Clients in Ihrer Umgebung verwendet haben. Die eindeutige Druckerkennung ist eine Kombination aus dem Namen des Druckertreibers (nicht Druckername), dem Standort des Druckers und der Schnittstelle.

„Einen Drucker nutzen“ bedeutet, dass innerhalb der Automation-Konfiguration eine der im Folgenden aufgeführten Aktionen ausgeführt wurde:

- Etikett drucken
- Daten an Drucker senden
- Etikettenvorschau
- Druckparameter festlegen

Jede dieser Aktionen zeigt an, dass ein Drucker genutzt wurde. Der jeweilige Drucker wird zur Liste genutzter Drucker hinzugefügt und verbleibt nach der letzten Nutzung 7 Tage lang dort. Um einen Drucker aus der Liste zu entfernen, müssen Sie ihn lediglich 7 Tage lang nicht verwenden, woraufhin er automatisch entfernt wird. In der Software wird die Information **Zuletzt verwendet** angezeigt, sodass Sie wissen, wenn die 7 Tage für den Drucker verstrichen sind. Sie können einen Druckerplatz an einen bestimmten Drucker binden, indem Sie in das Kontrollkästchen **Reserviert** klicken. Eine Reservierung sorgt dafür, dass ein Drucker jederzeit verfügbar ist – selbst wenn er mehr als 7 Tage nicht genutzt wurde.

WARNUNG

Wenn Sie die Anzahl an Druckerplätzen im Rahmen Ihrer Lizenz überschreiten, können Sie die Software 30 Tage lang in einem Übergangszeitraum nutzen. In diesem Modus wird die Anzahl der erlaubten Drucker vorübergehend auf die doppelte Anzahl der Druckerplätze angehoben, die gemäß Ihrer Lizenz zugelassen sind.

Der Übergangsmodus gibt Ihnen ausreichend Zeit, um Lizenzprobleme zu lösen, ohne dass es zu Ausfällen beim Drucken oder beim Erstellen von Etiketten kommt. Eine Überschreitung der Anzahl von erlaubten Drucker ist meistens auf ersetzte Drucker in Ihrer Umgebung zurückzuführen. Sie tritt ein, wenn alte und neue Drucker gleichzeitig verwendet werden oder wenn Sie neue Drucker hinzufügen. Wenn Sie den Lizenzkonflikt nicht innerhalb des Übergangszeitraums lösen, wird die Anzahl der erlaubten Drucker wieder auf die erworbene Anzahl von erworbenen Druckerplätzen zurückgesetzt, angefangen beim zuletzt verwendeten Drucker in der Liste.

TIPP

Um mehr über die NiceLabel 10 Lizenzierung zu erfahren, lesen Sie das Dokument zu diesem Thema– NiceLabel 10 Lizenzierung.

9.15. Im Dienstmodus ausführen

NiceLabel Automation wird als Windows-Dienst ausgeführt und wurde so konzipiert, dass bei der Verarbeitung von Daten und der Ausführung von Aktionen keine Benutzereingriffe nötig sind. Der Dienst ist dafür konfiguriert zu starten, wenn das Betriebssystem hochgefahren wird, und bleibt im Hintergrund aktiv, solange Windows ausgeführt wird. NiceLabel Automation speichert die Liste aller geladenen Konfigurationen und aktiven Trigger. Der letzte bekannte Status wird beim Neustart des Servers automatisch wiederhergestellt.

Der Dienst wird mit den Berechtigungen des Benutzerkontos ausgeführt, die bei der Installation ausgewählt wurden. Der Dienst übernimmt sämtliche Zugriffsrechte dieses Benutzerkontos, einschließlich solcher für freigegebene Ressourcen im Netzwerk wie Netzlaufwerke und Druckertreiber. Verwenden Sie daher das Konto eines vorhandenen Benutzers mit ausreichenden Berechtigungen, oder – noch besser – erstellen Sie ein eigenes Konto für NiceLabel Automation.

Sie können den Dienst verwalten, indem Sie die „Dienste“ in der Windows-Systemsteuerung starten. In neueren Windows-Betriebssystemen ist dies auch auf der Registerkarte „Dienste“ im Windows Task-Manager möglich. Sie können „Dienste“ verwenden, um folgende Aufgaben auszuführen:

- Den Dienst starten und anhalten.
- Das Konto ändern, unter dem sich der Dienst anmeldet.

Empfohlene Konfiguration des Benutzerkontos für den Dienst

- Nach Möglichkeit sollte der Dienst nicht unter dem lokalen Systemkonto ausgeführt werden. Dabei handelt es sich um ein vordefiniertes lokales Windows-Konto mit umfassenden Zugriffsrechten für den lokalen Computer, das aber für gewöhnlich keinerlei Zugriffsrechte für Netzwerkressourcen hat.NiceLabel Automation erfordert uneingeschränkten Zugriff auf den `%temp%`-Ordner des Kontos, welcher eventuell für das lokale Systemkonto nicht verfügbar ist.
- Wenn Sie ein **neues Benutzerkonto** für den NiceLabel Automation-Dienst erstellen, sollten Sie sich mit diesem neuen Benutzer mindestens einmal bei Windows anmelden. Auf diese Weise stellen Sie sicher, dass das Benutzerkonto vollständig erstellt wird. Z. B. wird der temporäre Ordner `%temp%` erst erstellt, wenn Sie sich anmelden.
- Deaktivieren Sie die Anforderung zur regelmäßigen Änderung des Passworts für dieses Benutzerkonto.
- Stellen Sie sicher, dass dieses Konto die Berechtigung zur **Anmeldung als Dienst** hat.
- Führen Sie den Dienst im x64-(64-Bit-)Modus aus.

Zugriff auf Ressourcen

NiceLabel Automation übernimmt alle Zugriffsrechte von dem Windows-Benutzerkonto, auf dem der Dienst ausgeführt wird. Der Dienst führt alle Aktionen unter diesem Kontonamen aus. Etiketten können geöffnet werden, wenn das Konto Berechtigung zum Zugriff auf die jeweiligen Dateien hat. Etiketten können gedruckt werden, wenn das Konto Zugriff auf den Druckertreiber hat.

Wenn Sie innerhalb des Dokumentenspeichers ein Revisionskontrollsystem und Genehmigungsschritte verwenden, müssen Sie das Dienst-Benutzerkonto zum Mitglied des „Nur-Drucken“-Profils machen, zum Beispiel **Bediener**. Danach müssen Sie dem Profil Zugriffsrechte für einen bestimmten Ordner zuweisen (**Nur-Lesen**-Modus oder Betreiber-Profil). So stellen Sie sicher, dass NiceLabel Automation nur genehmigte Etiketten verwendet, und keine Vorlagen.

Weitere Informationen finden Sie im Abschnitt Zugriff auf freigegebene Ressourcen im Netzwerk.

Dienstmodus: 32-Bit gegenüber 64-Bit

NiceLabel Automation kann auf 32-Bit-(x86-) und 64-Bit-(x64-)Systemen nativ ausgeführt werden. Der Ausführungsmodus wird vom Windows-Betriebssystem automatisch bestimmt.NiceLabel Automation wird auf 64-Bit-Windows-Systemen im 64-Bit-Modus und auf 32-Bit-Windows-Systemen im 32-Bit-Modus ausgeführt.

- **Drucken:** Die Ausführung von Automation als 64-Bit-Prozess hat einige Vorteile, z. B. direkte Kommunikation mit dem 64-Bit-Druckerspooler-Dienst unter 64-Bit-Windows. Dadurch wird das bekannte Problem vermieden, das durch SPLWOW64.EXE verursacht wird, eine Middleware, über die 32-Bit-Anwendungen auf den 64-Bit-Druckerspooler-Dienst zugreifen können.
- **Datenbankzugriff:** Die Ausführung des NiceLabel Automation-Dienstes als 64-Bit-Prozess erfordert, dass die 64-Bit-Version der Datenbanktreiber auf die Daten zugreifen kann. Weitere Informationen finden Sie im Abschnitt Zugriff auf Datenbanken.

ANMERKUNG

Falls Sie keine 64-Bit-Datenbanktreiber für Ihre Datenbank haben, können Sie NiceLabel Automation nicht im 64-Bit-Modus nutzen. Sie müssen es auf einem 32-Bit-System installieren oder die Ausführung im 32-Bit-Modus erzwingen.

x86-Betriebsmodus unter Windows x64 erzwingen
Es kann Gründe geben, NiceLabel Automation als 32-Bit-Anwendung unter 64-Bit-Windows auszuführen.

So erzwingen Sie den x86-Modus in NiceLabel Automation unter Windows x64:

- Wählen Sie Start > Ausführen.
- Geben Sie **regedit** ein und drücken Sie die Eingabetaste.
- Navigieren Sie zum Schlüssel

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\NiceLabelAutomationService2017
```

- Ändern Sie den Dateinamen in NiceLabelAutomationService10.x86.exe, aber behalten Sie den vorhandenen Pfad bei.
- Starten Sie den NiceLabel Automation-Dienst neu.

> **WARNUNG**
> Es empfiehlt sich nicht, den NiceLabel Automation-Dienstmodus zu ändern. Wenn Sie dennoch entscheiden, ihn zu ändern, sollten Sie auf jeden Fall einen umfassenden Trigger-Test durchführen, bevor Sie die Konfiguration in einer Produktionsumgebung bereitstellen.

9.16. Suchreihenfolge für angeforderte Dateien

Wenn NiceLabel Automation eine angegebene Etiketten- oder Bilddatei lädt, sucht es an verschiedenen zuvor definierten Orten nach ihr.

NiceLabel Automation geht dabei in der folgenden Reihenfolge vor:

1. Prüft, ob die Datei an dem durch die Aktion vorgegebenen Speicherort vorhanden ist.
2. Prüft, ob die Datei im selben Ordner wie die Konfigurationsdatei (.MISX) vorhanden ist.
3. Prüft, ob die Etikettendatei im Ordner .\Labels (bzw., für Grafikdateien, im Ordner .\Graphics) vorhanden ist.
4. Prüft, ob die Etikettendatei im Ordner ..\Labels (bzw., für Grafikdateien, im Ordner ..\Graphics) vorhanden ist.
5. Prüft, ob die Datei im globalen Etikettenordner (bzw., für Grafikdateien, im Grafikordner) vorhanden ist, der in den Optionen konfiguriert wurde.

Falls die Datei an keinem dieser Orte auffindbar ist, schlägt die Aktion fehl und ein Fehler wird gemeldet.

9.17. Zugriff auf Ihre Trigger sichern

Bestimmte Implementierungen erfordern sicheren Zugriff auf die Trigger. NiceLabel Automation ermöglicht es Ihnen, Sicherheitsmaßnahmen zu aktivieren, mit denen der Trigger-Zugriff nur von vertrauenswürdigen Netzwerkgeräten aus erfolgen kann. Die Sicherheitskonfiguration hängt von der Art des Triggers ab. Einige Trigger-Typen ermöglichen ohne weitere Schritte die Konfiguration des sicheren Zugriffs. Für alle Trigger, die auf dem TCP/IP-Protokoll basieren, können Sie außerdem alle Details innerhalb der Windows-Firewall definieren.

Firewall konfigurieren

Wenn Sie TCP/IP-basierte Trigger wie TCP/IP Server Trigger, HTTP Server Trigger oder Webdienst-Trigger verwenden, müssen Sie externen Anwendungen die Verbindung mit den Triggern erlauben. Jeder Trigger wird im NiceLabel Automation-Dienst ausgeführt; der Zugriff auf diesen Dienst wird von der Windows-Firewall reguliert.

ANMERKUNG

Standardmäßig ist die Windows-Firewall so konfiguriert, dass alle eingehenden Verbindungen zum NiceLabel Automation-Dienst erlaubt werden. Dies erleichtert Ihnen das Konfigurieren und Testen von Triggern, kann Sie aber andererseits angreifbar für unbefugten Zugriff machen.

Wenn die NiceLabel Automation-Implementierung in Ihrem Unternehmen strengen Sicherheitsrichtlinien unterliegt, müssen Sie die Firewall entsprechend diesen Vorgaben aktualisieren.

Zum Beispiel:

- Sie können die Firewall so anpassen, dass sie nur eingehenden Datenverkehr von bekannten Quellen akzeptiert.
- Sie können eingehende Daten nur über vordefinierte Ports zulassen.
- Sie können ausschließlich Verbindungen von bestimmten Benutzern zulassen.
- Sie können definieren, an welchen Schnittstellen Sie eingehende Verbindungen zulassen wollen.

Um Änderungen an der Windows-Firewall vorzunehmen, öffnen Sie die Verwaltungskonsole **Windows-Firewall mit erweiterter Sicherheit** unter **Systemsteuerung** > **System und Sicherheit** > **Windows-Firewall** > **Erweiterte Einstellungen**.

ANMERKUNG

Wenn NiceLabel Automation mit NiceLabel Control Center-Produkten verbunden ist, stellen Sie sicher, dass Sie eingehende Verbindungen an Port **56415/TCP aktivieren.** Wenn Sie diesen Port schließen, können Sie NiceLabel Automation nicht über die Systemsteuerung verwalten.

Zugriff auf Basis von Dateizugriffsrechten erlauben

Der Dateitrigger wird bei Zeitstempel-Änderungsereignissen in der/den überwachten Datei(en) ausgelöst. Sie müssen die Trigger-Dateien in einem Ordner ablegen, auf den der NiceLabel Automation-Dienst zugreifen kann. Das Benutzerkonto, unter dem der Dienst ausgeführt wird, muss Zugriffsrechte für die Dateien haben. Gleichzeitig legen Zugriffsrechte für den Speicherort fest, welche Benutzer und welche Anwendungen die Trigger-Datei speichern können. Sie sollten die Zugriffsrechte so konfigurieren, dass nur autorisierte Benutzer Dateien speichern können.

Zugriff auf Basis von IP-Adresse und Hostnamen erlauben

Sie können den Zugriff auf den TCP/IP-Server-Trigger mithilfe von zwei Listen mit IP-Adressen und Hostnamen schützen.

- Die erste Liste, **„Verbindungen von den folgenden Hosts zulassen“**, enthält IP-Adressen oder Hostnamen von Geräten, die Daten an den Trigger senden können. Wenn die IP-Adresse eines Geräts hier aufgelistet ist, kann dieses Gerät Daten an den Trigger senden.
- Die zweite Liste, **„Verbindungen von den folgenden Hosts nicht zulassen“**, enthält IP-Adressen oder Hostnamen, die keine Daten senden dürfen. Wenn die IP-Adresse eines Geräts hier aufgelistet ist, kann dieses Gerät keine Daten an den Trigger senden.

Zugriff auf Basis von Benutzernamen und Passwörtern erlauben

Sie können den Zugriff auf den HTTP-Server-Trigger schützen, indem Sie die Benutzerauthentifizierung aktivieren. Wenn sie aktiviert ist, muss jede HTTP-Anfrage, die an den HTTP-Server-Trigger gesendet wird, die Kombination aus **„Benutzernamen und Passwort“** enthalten, die der festgelegten Kombination entspricht.

Zugriff auf Basis von Anwendungsgruppen-Mitgliedschaft erlauben

Sie können den Zugriff auf den HTTP Server Trigger schützen, indem Sie Benutzer zu einer Anwendungsgruppe in Control Center hinzufügen. Wenn diese Option aktiviert ist, können nur authentifizierte Mitglieder dieser Gruppe auf den Trigger zugreifen.

9.18. Sitzungsdruck

Der Sitzungsdruck ermöglicht es Ihnen, mehrere Etiketten anhand eines einzigen Druckauftrags zu drucken. Wenn der Sitzungsdruck aktiviert ist, empfängt, verarbeitet und druckt der Drucker alle Etiketten im Druckauftrag auf einmal. Aufgrund dieses gebündelten Prozesses steigt die Druckgeschwindigkeit.

TIPP

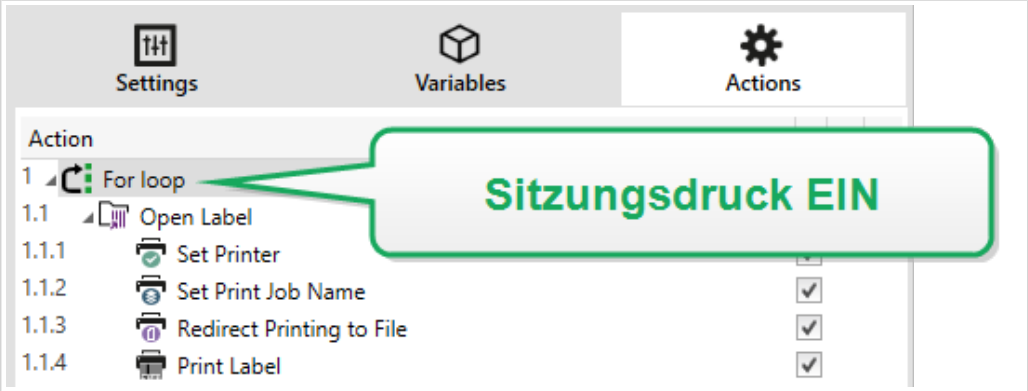
Sitzungsdruck dient als Alternative zum normalen Drucken, bei dem jedes Etikett als separater Druckauftrag an den Drucker gesendet wird.

ANMERKUNG

Automation aktiviert den Sitzungsdruck automatisch auf Basis der Konfiguration von Aktionen.

Wie wird der Sitzungsdruck gestartet?

Der Sitzungsdruck startet automatisch, wenn die Aktionen FOR Schleife, Für jeden Datensatz oder Für jede Zeile im Workflow vorhanden sind. In solchen Fällen initiiert die geschachtelte Aktion Etikett drucken den Sitzungsdruck automatisch. Das bedeutet, dass die Druckaktionen für alle Elemente in der Schleife in einem einzigen Druckauftrag enthalten sind.



Wie wird der Sitzungsdruck beendet?
Jede Drucksitzung endet entweder mit einer abgeschlossenen Schleife oder der Aktion Etikett drucken in Verbindung mit mindestens einer der folgenden Bedingungen:

- Der Drucker wird geändert. Wenn Sie anhand der Aktion Drucker einstellen einen anderen Drucker auswählen, endet der Sitzungsdruck.
- Die Druckerschnittstelle wird geändert. Wenn Sie den Druckauftrag anhand der Aktion Druck an Datei umleiten, wird der Sitzungsdruck beendet.
- Das Etikett wird geändert. Wenn Sie anhand der Aktion Etikett öffnen ein anderes zu druckendes Etikett auswählen, wird der Sitzungsdruck beendet.
- Es wird ein benutzerdefinierter Befehl gesendet, der den Sitzungsdruck beendet. Wenn Sie den Befehl `SESSIONEND` anhand der Aktion Benutzerdefinierte Befehle senden senden, wird der Sitzungsdruck beendet.

  ANMERKUNG
  `SESSIONEND` Muss als einziges Element der Aktion Benutzerdefinierte Befehle senden gesendet werden. Wenn Sie zusätzliche Befehle senden möchten, verwenden Sie separate „Benutzerdefinierte Befehle senden“-Aktionen.

ANMERKUNG
Bei komplexeren Konfigurationen sind möglicherweise mehrere Schleifen ineinander geschachtelt. In solchen Fällen wird der Sitzungsdruck beendet, wenn die äußerste übergeordnete Schleife endet.

9.19. Tipps und Tricks zur Nutzung von Variablen in Aktionen

Folgen sie den folgenden Empfehlungen, wenn Sie Variablen in NiceLabel Automation verwenden.

- **Schließen Sie Variablen in eckige Klammern ein.** Wenn Sie Variablen haben, deren Namen Leerzeichen enthalten, und in Aktionen auf Variablen verweisen, z. B. in SQL-Anweisung ausführen oder Skript ausführen, müssen Sie die Variablen in eckigen Klammern einschließen, z. B. **`[Produkt Name]`**. Außerdem sollten Sie eckige Klammern verwenden, falls Variablennamen mit reservierten Namen identisch sind, beispielsweise in der SQL-Anweisung.
- **Stellen Sie dem Variablennamen einen Doppelpunkt voran.** Um in der Aktion SQL-Anweisung ausführen oder in Datenbank-Trigger auf Variablen zu verweisen, müssen Sie dem Variablennamen einen Doppelpunkt (`:`) voranstellen, z. B. **`:[Produkt ID]`**. Der SQL-Parser fasst die jeweiligen Daten so als Variablenwert auf.

```
SELECT * FROM MyTable WHERE ID = :[ProductID]
```

- **Werte zwecks Berechnung in Ganzzahlen konvertieren.** Wenn Sie eine numerische Kalkulation mit Variablen durchführen wollen, müssen Sie sicherstellen, dass Sie die Variablenwerte in Ganzzahlen konvertieren. Die Definition von Variablen als rein numerisch beschränkt die Zeichen, die als Wert akzeptiert werden, ändert jedoch nicht den Variablentyp.NiceLabel Automation behandelt alle Variablen mit dem Typ Zeichenfolge. In VBScript würden Sie die Funktion **`CInt()`** nutzen.
- **Legen Sie Standard-/Startwerte für Skripte fest.** Wenn Sie Variablen in Aktionen verwenden, achten Sie darauf, dass diesen Variablen ein Standardwert zugewiesen ist, da ansonsten die Skriptprüfung fehlschlagen könnte. Sie können Standardwerte in den Variableneigenschaften oder in einem Skript festlegen (und wieder entfernen, nachdem Sie das Skript getestet haben).

9.20. Verfolgungsmodus

Standardmäßig speichert NiceLabel Automation Ereignisse in der Protokolldatenbank. Dies beinhaltet das Erheben von übergeordneten Daten, z. B.:

- Protokollieren des Ausführens von Aktionen
- Protokollieren der Filterausführung
- Protokollieren von Trigger-Statusaktualisierungen

Weitere Informationen finden Sie im Abschnitt Ereignisprotokoll-Optionen im Benutzerhandbuch.

Die Standard-Protokollierung zeichnet jedoch die tiefer gehenden Detailinformationen nicht auf. Wenn Fehlerbehebung auf tieferer Ebene im Code notwendig ist, müssen Sie den Verfolgungsmodus aktivieren. In diesem Modus protokolliert NiceLabel Automation Details zu allen internen Ausführungen, die während der Trigger-Verarbeitung stattfinden. Der Verfolgungsmodus sollte nur während der Problembehebung aktiviert werden, um Daten zu erheben, und im normalen Betrieb wieder deaktiviert werden.

WARNUNG

Der Verfolgungsmodus verlangsamt die Verarbeitung. Er sollte nur genutzt werden, wenn dies vom technischen NiceLabel Support-Team empfohlen wird.

Aktivierung des Verfolgungsmodus

Um den Verfolgungsmodus zu aktivieren, tun Sie Folgendes:

1. Navigieren Sie zum NiceLabel Automation-Systemordner.

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10
```

2. Erstellen Sie eine Sicherungskopie der Datei **`product.config`**.
3. Öffnen Sie die Datei **`product.config`** in einem Text-Editor. Die Datei hat eine XML-Struktur.
4. Fügen Sie der Datei das Element **`Common/Diagnostics/Tracing/Enabled`** hinzu und weisen sie ihm den Wert **True** zu.
   Die Datei sollte den folgenden Inhalt haben:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <Common>
        <Diagnostics>
            <Tracing>
                <Enabled>True</Enabled>
                <Folder>c:\Troubleshooting\TracingLogs</Folder>
            </Tracing>
        </Diagnostics>
    </Common>
...
</configuration>
```

5. Nach Speichern der Datei wendet der NiceLabel Automation-Dienst die Einstellung automatisch an.
6. Verfolgungsdateien (Dateierweiterung *.LOG) werden standardmäßig im selben Systemordner gespeichert.
   Um den Protokollordner zu übergehen, verwenden Sie die Datei **`product.config`**. Geben Sie den benutzerdefinierten Protokollordner im **`Folder`**-Element an. Dieses Element ist optional.
7. Um die Aktivierung des Verfolgungsmodus zu bestätigen, starten Sie den Automation Manager. Nach Aktivierung des Verfolgungsmodus wird der Text **Verfolgung wurde aktiviert** im Mitteilungsbereich über der Trigger-Liste angezeigt.

9.21. Informationen zu Druckereinstellungen und DEVMODE

ANMERKUNG

Die DEVMODE-Datenstruktur ist Teil der GDI Print API-Struktur in Windows. Dieser Abschnitt enthält hochgradig technische Inhalte, die nur für spezielle Anforderungen relevant sind.

Wenn Sie Etiketten mit NiceLabel Software drucken (oder auch ein Dokument in einer Windows-Anwendung), liest die druckende Anwendung die im Druckertreiber definierten Druckereinstellungen und wendet sie auf den Druckauftrag an. Dasselbe Etikett kann einfach durch Auswahl eines anderen Druckertreibers auf verschiedenen Druckern gedruckt werden. Dabei werden jedes Mal die Druckereinstellungen eines neu ausgewählten Druckers auf das Etikett angewandt.

Das Drucken eines Textdokuments anhand von zwei verschiedenen Laserdruckern führt normalerweise zum selben oder zumindest zu einem ähnlichen Ergebnis. Das Drucken von Etiketten anhand von zwei verschiedenen Etikettendruckern kann zu äußerst uneinheitlichen Ergebnissen führen. Um vergleichbare Ergebnisse zu erzielen, sind für dieselbe Etikettendatei eventuell zusätzliche Druckertreiber-Einstellungen erforderlich, z. B. Anpassungen des Versatzes, der Geschwindigkeit und der Drucktemperatur. Auch NiceLabel wendet Druckereinstellungen auf jeden Druckvorgang an. Standardmäßig werden Druckereinstellungen für den ausgewählten Drucker in der Etikettendatei gespeichert.

Was ist DEVMODE?

DEVMODE ist eine Windows-Struktur, in der Druckereinstellungen gespeichert sind (Initialisierungs- und Umgebungsinformationen zu einem Drucker). Sie beinhaltet zwei Teile: öffentlich und privat. Der öffentliche Teil enthält Daten, die für alle Drucker gelten. Der private Teil enthält Daten, die speziell für einen bestimmten Drucker gelten. Der private Teil kann eine variable Länge aufweisen und enthält alle spezifischen Hersteller-bezogenen Einstellungen.

- **Öffentlicher Teil:** In diesem Teil sind die allgemeinen Einstellungen codiert, die im Druckertreibermodell verfügbar gemacht werden, z. B. Druckername, Treiberversion, Papiergröße, Ausrichtung, Farbe, Duplex und ähnliche. Der öffentliche Teil bleibt für alle Druckertreiber identisch und bietet keine Unterstützung für die speziellen Einstellungen von Etikettendruckern (Thermodrucker, industrielle Tintenstrahldrucker, Lasergravur-Geräte).
- **Privater Teil:** In diesem Teil sind Einstellungen codiert, die im öffentlichen Teil nicht enthalten sind. NiceLabel-Druckertreiber nutzen diesen Teil zum Speichern der Druckermodell-spezifischen Daten, z. B. Druckgeschwindigkeit, Temperatureinstellungen, Versätze, Druckmodus, Medientyp, Sensoren, Abschneider, Grafik-Codierung, RFID-Unterstützung und ähnliches. Die Datenstruktur innerhalb des privaten Teils des DEVMODE ist ein Strom aus Binärdaten, der von Treiberentwicklern festgelegt wird.

DEVMODE-Änderungen

Die DEVMODE-Datenstruktur ist in der Windows-Registrierung gespeichert. Es gibt zwei Kopien der Struktur: Standard-Druckereinstellungen und benutzerspezifische Druckereinstellungen. Sie können die DEVMODE (Druckereinstellungen) abändern, indem Sie die Parameter im Druckertreiber ändern. Die ersten beiden Optionen sind Windows-bezogen, die dritte steht in der NiceLabel Software zur Verfügung.

- **Standard-Druckereinstellungen:** Diese Einstellungen sind unter **Druckereigenschaften** > **Erweitert-Registerkarte** > **Druckstandards**.
- **Benutzerspezifische Einstellungen:** Diese Einstellungen werden für jeden Benutzer in dessen HKEY_CURRENT_USER-Registrierungsschlüssel gespeichert. Standardmäßig werden benutzerspezifische Einstellungen von den Drucker-Standardeinstellungen übernommen. Die benutzerspezifischen Einstellungen sind unter **Druckereigenschaften > Einstellungen** definiert. Alle hier vorgenommenen Änderungen betreffen nur den aktuellen Benutzer.
- **Etiketten-spezifische Einstellungen:** Der Etikettenautor, der die NiceLabel Software nutzt, kann die DEVMODE in das Etikett integrieren. Auf diese Weise können Druckereinstellungen zwischen Rechnern ausgetauscht werden. Beim Kopieren des Etiketts werden Druckereinstellungen einfach mit übernommen. Um Druckereinstellungen in ein Etikett einzubetten, aktivieren Sie die Option **Im Etikett gespeicherte benutzerdefinierte Druckereinstellungen verwenden** anhand der *Dokumenteigenschaften* in Designer Pro. Sie können die auf dem Etikett eingebetteten Druckereinstellungen ändern, indem Sie den *Drucker*-Bereich in den *Dokumenteigenschaften* auswählen.

Benutzerdefinierte DEVMODE-Struktur auf den Ausdruck anwenden

In NiceLabel Automation können Sie eine Etikettendatei öffnen und eine benutzerdefinierte DEVMODE darauf anwenden. Beim Drucken eines Etiketts wird sein Design aus der .NLBL-Datei entnommen, und die DEVMODE wendet die spezifische, druckerbezogene Formatierung an. Auf diese Weise ist es möglich, nur ein einzelnes Etiketten-Master zu verwenden. In diesem Fall ist der Ausdruck des Etiketts auf jedem Drucker identisch, da jeweils die optimalen Druckereinstellungen angewandt werden.

Um eine benutzerdefinierte DEVMODE auf ein Etikett anzuwenden, haben Sie zwei Optionen:

1. Anhand einer Aktion, genauer gesagt anhand des Parameters **Druckereinstellungen**.
2. Anhand der JOB-Befehlsdatei, genauer gesagt anhand des Befehls **SETPRINTPARAM** mit dem Parameter **PRINTERSETTINGS**. Weitere Informationen finden Sie im Abschnitt Benutzerdefinierte Befehle verwenden des Benutzerhandbuchs.

9.22. Dasselbe Benutzerkonto zur Konfiguration und Ausführung von Triggern verwenden

Der NiceLabel Automation-Dienst wird immer mit den Anmeldedaten des Benutzerkontos ausgeführt, das für den Dienst konfiguriert wurde. Automation Builder jedoch wird immer mit den Anmeldedaten des

aktuell angemeldeten Benutzers ausgeführt. Die Anmeldedaten für das Dienstkonto und das aktuell angemeldete Konto können abweichen.

Während Sie im Automation Builder problemlos eine Vorschau von Triggern anzeigen können, könnte der Dienst eine Fehlermeldung aufgrund falscher Anmeldedaten ausgeben. Während der aktuell angemeldete Benutzer Berechtigung zum Zugriff auf Ordner und Drucker hat, gilt dies für das vom Dienst genutzte Benutzerkonto eventuell nicht.

Sie können die Ausführung von Triggern in Automation Builder anhand derselben Anmeldedaten testen, die auch der Dienst nutzt. Führen Sie Automation Builder zu diesem Zweck unter demselben Benutzerkonto aus, das für den Dienst definiert ist.

Um Automation Builder unter einem anderen Benutzerkonto auszuführen, tun Sie Folgendes:

1. Drücken und halten Sie die **Umschalttaste** und klicken Sie mit der **rechten Maustaste** auf das Automation Builder-Symbol.
2. Wählen Sie **Als anderer Benutzer ausführen**.
3. Geben Sie die Anmeldedaten für den Benutzer ein, der für den NiceLabel Automation-Dienst verwendet wird.
4. Klicken Sie auf **OK**.

Wenn Sie vorhaben, Automation Builder häufig mit Anmeldedaten des anderen Benutzerkontos auszuführen, verwenden Sie das in Windows bereitgestellte Befehlszeilen-Hilfsprogramm **RUNAS**. Verwenden Sie die Switches `/user`, um das Benutzerkonto festzulegen, und `/savecred`. Durch letzteren Switch müssen Sie das Passwort nur einmal eingeben; danach wird es automatisch abgerufen.