

# NiceLabel Automation 2017 Benutzerhandbuch

Rev-1702 ©NiceLabel 2017.

# 1 Inhalt

<b>1 Inhalt</b> .....	<b>2</b>
<b>2 Willkommen bei NiceLabel Automation</b> .....	<b>8</b>
<b>3 Typografische Konventionen</b> .....	<b>10</b>
<b>4 Anwendung einrichten</b> .....	<b>11</b>
4.1 Architektur .....	11
4.2 Systemanforderungen .....	11
4.3 Installation .....	12
4.4 Aktivierung .....	13
4.5 Testmodus .....	13
4.6 Datei-Tab .....	14
4.6.1 Öffnen .....	14
4.6.2 Kompatibilität mit NiceWatch-Produkten .....	14
4.6.3 Speichern .....	16
4.6.4 Speichern unter .....	16
4.6.5 Optionen .....	16
4.6.6 Über .....	21
<b>5 Informationen zu Filtern</b> .....	<b>23</b>
5.1 Informationen zu Filtern .....	23
5.2 Filter für strukturierten Text konfigurieren .....	24
5.2.1 Filter für strukturierten Text .....	24
5.2.2 Felder definieren .....	25
5.2.3 Dynamische Struktur aktivieren .....	27
5.3 Filter für unstrukturierte Daten konfigurieren .....	29
5.3.1 Filter für unstrukturierte Daten .....	29
5.3.2 Felder definieren .....	31
5.3.3 Unterbereiche definieren .....	34
5.3.4 Zuweisungsbereiche definieren .....	36
5.4 XML-Filter konfigurieren .....	38
5.4.1 XML-Filter .....	38
5.4.2 XML-Felder definieren .....	39
5.4.3 Sich wiederholende Elemente definieren .....	41

5.4.4 XML-Zuweisungsbereich definieren .....	43
5.5 Etiketten- und Druckernamen anhand von Eingabedaten einstellen .....	45
<b>6 Trigger konfigurieren .....</b>	<b>47</b>
6.1 Informationen zu Triggern .....	47
6.2 Trigger definieren .....	49
6.2.1 Dateitrigger .....	49
6.2.2 Trigger für serielle Schnittstelle .....	53
6.2.3 Datenbank-Trigger .....	55
6.2.4 TCP/IP Server Trigger .....	62
6.2.5 HTTP Server Trigger .....	66
6.2.6 Webdienst-Trigger .....	72
6.3 Variablen verwenden .....	82
6.3.1 Variablen .....	82
6.3.2 Zusammengesetzte Werte verwenden .....	83
6.3.3 Interne Variablen .....	84
6.3.4 Globale Variablen .....	86
6.4 Aktionen verwenden .....	87
6.4.1 Aktionen .....	87
6.4.1.1 Aktionen definieren .....	87
6.4.1.2 Geschachtelte Aktionen .....	88
6.4.1.3 Ausführen von Aktionen .....	88
6.4.1.4 Bedingungsabhängige Aktionen .....	88
6.4.1.5 Aktionen mit Konfigurationsfehlern erkennen .....	89
6.4.1.6 Aktionen deaktivieren .....	89
6.4.1.7 Aktionen kopieren .....	90
6.4.1.8 In der Aktionsliste navigieren .....	90
6.4.1.9 Die Aktionen beschreiben .....	90
6.4.2 Allgemein .....	91
6.4.2.1 Etikett öffnen .....	91
6.4.2.2 Etikett drucken .....	93
6.4.2.3 Oracle XML-Befehlsdatei ausführen .....	96
6.4.2.4 SAP All XML-Befehlsdatei ausführen .....	98
6.4.2.5 Befehlsdatei ausführen .....	100
6.4.2.6 Benutzerdefinierte Befehle senden .....	102

6.4.3 Drucker .....	103
6.4.3.1 Drucker einstellen .....	103
6.4.3.2 Druckauftragsnamen festlegen .....	105
6.4.3.3 Druck an Datei umleiten .....	106
6.4.3.4 Druckparameter festlegen .....	109
6.4.3.5 Druckumleitung an PDF .....	112
6.4.3.6 Druckerstatus .....	114
6.4.3.7 Etikett im Drucker speichern .....	117
6.4.4 Variablen .....	119
6.4.4.1 Variable einstellen .....	119
6.4.4.2 Variable Daten speichern .....	120
6.4.4.3 Variable Daten laden .....	122
6.4.4.4 Zeichenfolgenmanipulation .....	124
6.4.5 Stapeldruck .....	127
6.4.5.1 FOR Schleife .....	127
6.4.5.2 Datenfilter verwenden .....	129
6.4.5.3 Für jeden Datensatz .....	131
6.4.6 Daten und Konnektivität .....	133
6.4.6.1 Dokument/Programm öffnen .....	133
6.4.6.2 Daten in Datei speichern .....	135
6.4.6.3 Daten aus Datei lesen .....	136
6.4.6.4 Datei löschen .....	139
6.4.6.5 SQL-Anweisung ausführen .....	140
6.4.6.6 Daten an TCP/IP-Port senden .....	144
6.4.6.7 Daten an serielle Schnittstelle senden .....	146
6.4.6.8 Daten von serieller Schnittstelle lesen .....	148
6.4.6.9 Daten an Drucker senden .....	150
6.4.6.10 HTTP-Anfrage .....	152
6.4.6.11 Webdienst .....	155
6.4.7 Sonstiges .....	158
6.4.7.1 Etiketteninformationen abrufen .....	158
6.4.7.2 Script ausführen .....	162
6.4.7.2.1 Skript-Editor .....	164
6.4.7.3 Meldung (Konfiguration) .....	165

6.4.7.4 Lizenz verifizieren .....	167
6.4.7.5 Testen .....	168
6.4.7.6 XML-Umwandlung .....	170
6.4.7.7 Gruppe .....	173
6.4.7.8 Ereignis protokollieren .....	174
6.4.7.9 Etikettenvorschau .....	176
6.4.7.10 Etikettenvariante erstellen .....	177
6.5 Trigger testen .....	180
6.5.1 Trigger testen .....	180
6.6 Trigger-Konfiguration vor Bearbeitung schützen .....	182
6.7 Firewall für Netzwerk-Trigger konfigurieren .....	183
6.8 Sicheres Übertragungsprotokoll (HTTPS) nutzen .....	184
<b>7 Trigger ausführen und verwalten .....</b>	<b>187</b>
7.1 Konfiguration anwenden .....	187
7.2 Ereignisprotokoll-Optionen .....	188
7.3 Trigger verwalten .....	188
7.4 Ereignisprotokoll verwenden .....	190
<b>8 Performance- und Feedback-Optionen .....</b>	<b>192</b>
8.1 Parallele Verarbeitung .....	192
8.2 Dateien zwischenspeichern .....	193
8.3 Fehlerhandhabung .....	195
8.4 Synchroner Druckmodus .....	197
8.5 Feedback zum Status von Druckaufträgen .....	198
8.6 Speichern/Abrufen-Druckmodus verwenden .....	200
8.7 Hochverfügbarkeits-(Failover-)Cluster .....	201
8.8 Lastausgleichs-Cluster .....	202
<b>9 Informationen zu Datenstrukturen .....</b>	<b>204</b>
9.1 Informationen zu Datenstrukturen .....	204
9.2 Binärdateien .....	204
9.3 Befehlsdateien .....	205
9.4 Zusammengesetzte CSV-Dateien .....	205
9.5 Altdateien .....	206

9.6 Textdatenbank .....	206
9.7 XML-Daten .....	207
<b>10 Referenz und Fehlerbehebung .....</b>	<b>210</b>
10.1 Typen von Befehlsdateien .....	210
10.1.1 Spezifikationen für Befehlsdateien .....	210
10.1.2 CSV-Befehlsdatei .....	210
10.1.3 JOB-Befehlsdatei .....	211
10.1.4 XML-Befehlsdatei .....	212
10.1.5 Oracle XML-Spezifikationen .....	216
10.1.6 SAP All XML-Spezifikationen .....	217
10.2 Benutzerdefinierte Befehle .....	218
10.2.1 Benutzerdefinierte Befehle verwenden .....	218
10.3 Zugriff auf freigegebene Ressourcen im Netzwerk .....	224
10.4 Dokumentenspeicher und Versionierung von Konfigurationsdateien .....	225
10.5 Zugriff auf Datenbanken .....	226
10.6 Automatisches Ersetzen von Schriften .....	226
10.7 Standardeinstellungen für Multi-Thread-Druck ändern .....	228
10.8 Kompatibilität mit NiceWatch-Produkten .....	229
10.9 Dienst mit Befehlszeilenparametern steuern .....	231
10.10 Ersetzen der Datenbank-Verbindungszeile .....	233
10.11 Eingabe von Sonderzeichen (Steuercodes) .....	234
10.12 Liste mit Steuercodes .....	235
10.13 Lizenzierung und Druckernutzung .....	236
10.14 Im Dienstmodus ausführen .....	237
10.15 Suchreihenfolge für die angeforderten Dateien .....	239
10.16 Zugriff auf Ihre Trigger sichern .....	239
10.17 Sitzungsdruck .....	241
10.18 Tipps und Tricks zur Nutzung von Variablen in Aktionen .....	242
10.19 Verfolgungsmodus .....	243
10.20 Informationen zu Druckereinstellungen und DEVMODE .....	244
10.21 Dasselbe Benutzerkonto zur Konfiguration und Ausführung von Triggern verwenden .....	246
<b>11 Beispiele .....</b>	<b>247</b>

11.1 Beispiele .....	247
<b>12 Technischer Support .....</b>	<b>248</b>
12.1 Online-Support .....	248

# 2 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, Lieferketten usw. So können alle Anwendungen in allen Abteilungen und Niederlassungen Ihres Unternehmens autorisierte Etiketten drucken.

NiceLabel Automation ermöglicht optimalen Etikettendruck auf Unternehmensebene, 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.

Der automatisierte Etikettendruck mit einer Trigger-basierten Anwendung dreht sich um drei Kernprozesse.

## Trigger

Trigger sind eine einfache, aber leistungsstarke Funktion, die Ihnen bei der Automatisierung von Aufgaben hilft. Grundsätzlich handelt es sich bei einem Trigger um eine Ursache-und-Wirkung-Aussage: Er führt eine bestimmte Aktion aus, sobald ein bestimmtes Ereignis eintritt.

Es geht hier also um eine **WENN ... DANN**-Verarbeitung. Trigger eignen sich für Aufgaben, die Sie wiederholt ausführen.

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 ein, werden eine Änderung an einer Datei oder eingehende Daten verzeichnet, was wiederum den Etikettendruckprozess auslöst.

Im entsprechenden Kapitel finden Sie weitere Informationen zu den verschiedenen Arten von [Triggern](#):

- Datei-Trigger
- Schnittstellen-Trigger
- Datenbank-Trigger
- TCP/IP-Trigger
- HTTP-Trigger
- Web Service-Trigger

## Datenextraktion und -platzierung

Nachdem der Druckprozess ausgelöst wurde, extrahiert NiceLabel Automation Etikettendaten und fügt sie in die variablen Felder auf der Etikettenvorlage ein.



[Filter](#) für die Datenextraktion unterstützen:

- Strukturierte Textdateien
- Unstrukturierte Textdateien
- Verschiedene XML-Dateien
- Binäre Daten: Druckerersatz, Exporte aus alter Software, Daten von Hardware-Geräten usw.

### **Ausführen von Aktionen**

Nachdem die Daten den variablen Feldern auf dem Etikett zugeordnet wurden, führt NiceLabel Automation Aktionen aus. Zu den grundlegenden Operationen zählen die Aktionen **Etikett öffnen** und **Etikett drucken**, mit denen die extrahierten Daten auf das Etikett gedruckt werden. Darüber hinaus können die Daten auch an individuelle Orte gesendet werden, z. B. als Dateien an die Festplatte, an Webserver, Hardwaregeräte usw. Insgesamt können Sie aus über 30 verschiedenen Aktionen wählen.

Im entsprechenden Kapitel finden Sie weitere Informationen über erweiterte [Druckaktionen](#).

# 3 Typografische Konventionen

Text in **Fettdruck** verweist auf Menünamen und Schaltflächen.

Text in *Kursivdruck* bezieht sich auf Optionen, Bestätigungsaktionen wie „Nur lesen“ und Orte wie z. B. Verzeichnisse.

Text in <spitzen Klammern> verweist auf Tasten der PC-Tastatur, beispielsweise <Eingabe>.

Variablen erscheinen in [eckigen Klammern].

**HINWEIS:** Dies ist der Stil für Hinweise.

**BEISPIEL:** Dies ist der Stil für Beispiele.

Dies ist der Stil für optimale Vorgehensweisen.

**WARNUNG:** Dies ist der Stil für Warnungen.

**TIPP:** Dies ist der Stil für Tipps.

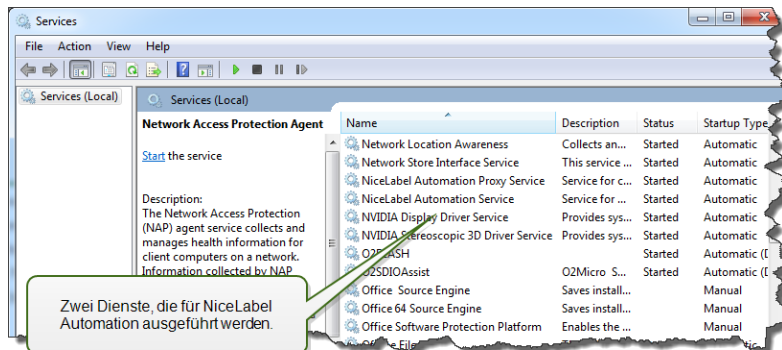
# 4 Anwendung einrichten

## 4.1 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.** Dies ist die Konfigurationsanwendung, mit der Entwickler Trigger, Filter und Aktionen erstellen, die ausgeführt werden, wenn Daten beim Trigger eingeht. Diese Anwendung wird immer im 32-Bit-Modus ausgeführt.
- **Automation Manager.** Dies ist die Verwaltungsanwendung, die verwendet wird, um die Ausführung von Triggern in Echtzeit zu überwachen und die Trigger zu starten/anzuhalten. Diese Anwendung wird immer im 32-Bit-Modus 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.



## 4.2 Systemanforderungen

- CPU: Intel- oder kompatibler Prozessor der x86-Reihe
- Arbeitsspeicher: 2 GB RAM oder mehr
- Festplatte: 1 GB an freiem Festplattenspeicher
- 32- oder 64-Bit-Version Windows Betriebssystem: Windows Server 2008 R2, Windows 7, Windows 8, Windows 8.1, Windows Server 2012, Windows Server 2012 R2, Windows 10, Windows Server 2016 (Windows Server Core und Windows Nano Server werden nicht

unterstützt)

- Microsoft .NET Framework Version 4.5
- Anzeige: Monitor mit 1366 × 768 oder höherer Auflösung
- Etiketten-Designer:
  - Empfohlen: NiceLabel V2017 (.NLBL-Dateiformat)
  - Mindestvoraussetzung: NiceLabel Pro V5.4 (.LBL-Dateiformat), wobei es jedoch zu Kompatibilitätsproblemen kommen kann
- Empfohlene Druckertreiber: NiceLabel Printer Drivers V5.1 oder neuer
- Uneingeschränkter Zugriff auf den Systemordner der Anwendung, wo Ereignisse in einer Datenbank protokolliert werden.

```
%PROGRAMDATA%\NiceLabel\NiceLabel 2017
```

- Uneingeschränkter Zugriff auf den %temp%-Ordner des Benutzeraccounts.

## 4.3 Installation

**HINWEIS:** Im Folgenden finden Sie die zusammengefasste Version des Installationsverfahrens. Weitere Informationen finden Sie im **Installationshandbuch**.

Bevor Sie mit der Installation beginnen, müssen Sie sicherstellen, dass Ihre Infrastruktur mit den [Systemanforderungen](#) kompatibel ist.

So installieren Sie NiceLabel Automation:

1. Legen Sie die NiceLabel-DVD ein.

Die Hauptmenü-Anwendung startet automatisch.

Ist dies nicht der Fall, doppelklicken Sie auf die Datei **START.EXE** auf der DVD.

2. Klicken Sie auf **NiceLabel-Produkte 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. Geben Sie auf jeden Fall einen existierenden Benutzernamen an, da der Dienst die Benutzerrechte dieses Benutzernamens übernehmen wird. Weitere Informationen finden Sie im Kapitel [Im Dienstmodus ausführen](#).

### Upgrade

Um ein Upgrade für NiceLabel Automation 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, wobei die vorhandenen Einstellungen

beibehalten werden. Während des Upgrades wird die Protokolldatenbank geleert.

**HINWEIS:**Zwei unterschiedliche Hauptversion des selben NiceLabel-Produktes sind gleichzeitig installiert.

## 4.4 Aktivierung

Sie müssen die NiceLabel Automation-Software aktivieren, damit die konfigurierten Trigger verarbeitet werden können. Der Aktivierungsvorgang erfordert eine Internetverbindung, vorzugsweise auf dem Rechner, auf dem die Software installiert wird. Die Aktivierung der Testlizenz erfolgt anhand desselben Verfahrens.

**HINWEIS:**Sie können die Software entweder in Automation Builder oder in Automation Manager aktivieren, wobei das Ergebnis dasselbe ist.

### Aktivierung in Automation Builder

1. Starten Sie **Automation Builder**.
2. Wählen Sie **Datei>Über>Ihre Lizenz aktivieren**.  
Der Aktivierungsassistent wird 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.

## 4.5 Testmodus

Im Testmodus können Sie NiceLabel Automation bis zu 30 Tage lang testen. Der Testmodus bietet denselben Funktionsumfang wie die lizenzierte Version. So können Sie das Produkt vor dem Kauf eingehend ausprobieren. In Automation Manager wird durchgehend eine Meldung mit einem Hinweis auf die Testversion und die verbleibende Anzahl von Tagen angezeigt. Nach Ablauf des Testzeitraums werden vom NiceLabel Automation-Dienst keine Trigger mehr verarbeitet. Der 30-Tage-Zeitraum beginnt am Tag der Installation.

**HINWEIS:**Sie können den Testzeitraum verlängern, indem Sie Ihren NiceLabel Händler kontaktieren und einen weiteren Testlizenzschlüssel anfordern. Sie müssen den Testlizenzschlüssel aktivieren. Weitere Informationen finden Sie im Abschnitt [Aktivierung](#).

## 4.6 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.

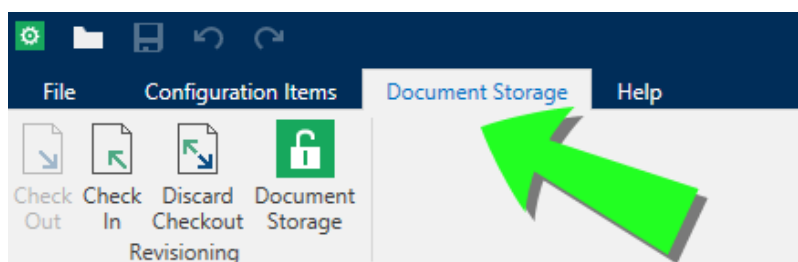
### 4.6.1 Öffnen

Mit dem Öffnen-Dialog können Sie vorhandene 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. Wenn für diesen Ort im Control Center Dokumentenversionierung aktiviert ist, wird eine zusätzliche Registerkarte geöffnet. 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.

### 4.6.2 Kompatibilität Mit NiceWatch-Produkten

NiceLabel Automation kann Trigger-Konfigurationen laden, die in einem der NiceWatch-Produkte definiert wurden. In den meisten Fällen können Sie die NiceWatch-Konfiguration in NiceLabel Automation ohne Anpassungen ausführen.

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 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.

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

- **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 in NiceWatch-Produkten stehen in NiceLabel Automation zur Verfügung. Einige sind überhaupt nicht verfügbar, während andere nur unterschiedlich konfiguriert werden. Wenn die MIS-Datei nicht unterstützte Funktionen enthält, wird Ihnen eine Liste dieser Funktionen angezeigt und sie werden aus der Konfiguration entfernt.

In diesem Fall müssen Sie die .MIS-Datei in Automation Builder öffnen und die Kompatibilitätsprobleme beheben. Sie müssen die Funktionen von NiceLabel Automation nutzen, um die entfernte Konfiguration neu zu erstellen.

- **Kompatibilität mit den Etikettenvorlagen.** Falls Ihre vorhandenen Etikettenvorlagen Funktionen enthalten, die von der Druck-Engine in NiceLabel Automation nicht unterstützt werden, sehen Sie eine Fehlermeldung im Log-Bereich. 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.

**HINWEIS:** Weitere Informationen über Kompatibilitätsprobleme mit NiceWatch und Etiketten-Designern finden Sie im [Knowledge Base-Artikel KB251](#).

### **NiceWatch-Konfiguration zur Bearbeitung öffnen**

Sie können die vorhandene NiceWatch-Konfiguration (.MIS-Datei) in Automation Builder öffnen und in Automation Builder bearbeiten. Sie können die Konfiguration nur im .MISX-Format öffnen.

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. Falls die Konfiguration nicht unterstützte Funktionen enthält, werden diese als Liste angezeigt. Sie werden aus der Konfiguration entfernt.

### 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. Im Automation Manager wird der Trigger für die ausgewählte Konfiguration angezeigt. Um den Trigger zu starten, wählen Sie ihn aus und klicken Sie auf die Schaltfläche **Start**.

**HINWEIS:**Falls ein Kompatibilitätsproblem mit der NiceWatch-Konfiguration besteht, müssen Sie sie in Automation Builder öffnen und umkonfigurieren.

## 4.6.3 Speichern

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

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

## 4.6.4 Speichern Unter

**Speichern unter** ermöglicht es Ihnen, die aktive Konfigurationsdatei unter Angabe eines Namens und Speicherorts zu speichern.

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

## 4.6.5 Optionen

Nutzen Sie die Einstellungen in diesem Dialogfeld, um die Anwendung anzupassen. Wählen Sie die Gruppe im linken Feld aus und konfigurieren Sie die Einstellungen dann im rechten Feld.

### Ordner

Sie können Standardordner zum Speichern der Etiketten, Eingabemasken, Datenbanken und Bilddateien auswählen. Der Standardspeicherort ist der Dokumente-Ordner der aktuellen Benutzers. In diesen Standardordnern sucht NiceLabel Automation immer dann nach Dateien,



wenn Sie nur den Dateinamen, aber keinen vollständigen Pfad angeben. Weitere Informationen zur Suchreihenfolge finden Sie im Abschnitt [Suchreihenfolge für die angeforderten Dateien](#).

Die Änderungen werden binnen einer Minute an den Dienst weitergegeben. Um Änderungen umgehend anzuwenden, können Sie den NiceLabel Automation-Dienst neu starten.

**HINWEIS:**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 zuerst 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

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

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

### Globale Variablen

Das Tab "Globale Variablen" ermöglicht es Ihnen, festzulegen, welcher Speicherort für globale Variablen verwendet werden soll:

- **Auf dem Server ([ControlCenter]) gespeicherte globale Variablen verwenden.** Legt das Control Center als Speicherort für globale Variablen fest.

**HINWEIS:**Diese Option wird verfügbar, wenn Sie die NiceLabel Label LMS-Lizenz verwenden.

- **In einer Datei (lokal oder in Freigabe) gespeicherte globale Variablen verwenden.** Legt einen lokalen oder im Netzwerk freigegebenen Ordner als Speicherort für globale Variablen fest. Geben Sie den genauen Pfad ein oder klicken Sie auf **Öffnen**, um die Datei zu finden.

### Control Center

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

**HINWEIS:**Dieses Tab ist nur bei aktivierter LMS-Lizenz 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.

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

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

- **Druckereignisse:** protokolliert die druckbezogenen Ereignisse vom Rechner.
- **Fehlerereignisse:** protokolliert alle gemeldeten Fehler.

**HINWEIS:** Standardmäßig werden Druck- und Fehlerereignisse in 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.

- **Druckauftragprotokoll auf dem Server aktivieren:** aktiviert die Protokollierung von Druckaufträgen.
- **Detaillierte Druckerkontrolle:** ermöglicht die Überwachung der Status, die vom verbundenen Drucker gemeldet werden.

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

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

## Druckernutzung

**HINWEIS:**Die Protokollierung der Druckernutzung ist auch für Lizenzen mit mehreren Druckerplätzen verfügbar.

Das Tab **Druckernutzung** zeigt die protokollierte Nutzung der installierten Drucker an. Die Druckernutzung gibt Ihnen Aufschluss über die Anzahl von Druckern, die in Ihrer Druckumgebung verwendet wurden.

Die Gruppe **Druckernutzungs-Informationen** zeigt an, wie viele der erlaubten Druckerschnittstellen durch den Druck auf mehreren Druckern verwendet werden.

- **Anzahl von Druckern im Rahmen der Lizenz.** Anzahl von Druckern, die mit der aktuellen Designer-Lizenz verwendet werden können.

- **Anzahl verwendeter Drucker in den letzten 7 Tagen.** Anzahl der Drucker, die innerhalb der letzten 7 Tage mit Designer verwendet wurden.

**TIPP:** Innerhalb von 7 Tagen ermöglicht die Designer-Lizenz nur die Nutzung der angegebenen Anzahl unterschiedlicher Drucker.

**WARNUNG:** Wenn die zulässige Anzahl von Druckern – wie von der Lizenz vorgegeben – überschritten wird, wird eine Warnung angezeigt. Nach Verdopplung der Anzahl erlaubter Drucker kann auf Druckern, die darüber hinaus hinzugefügt werden, nicht mehr gedruckt werden.

Die Druckstatus werden in mehreren Spalten angezeigt:

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

**HINWEIS:** 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.
- **Schnittstelle.** 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.

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

Die **Berechtigungen**-Gruppe 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 2017 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 dedizierte Thermo- oder Laser-Etikettendrucker und beschränken Sie das Drucken auf sie, um ununterbrochenes Drucken von Etiketten mit einer Mehrbenutzerlizenz zu garantieren.

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

1. Navigieren Sie zum Systemverzeichnis.

**BEISPIEL:** [[[Undefined variable Variables.Path-System-Designer]]]

2. Erstellen Sie eine Sicherungskopie der `product.config`-Datei.

3. Öffnen Sie `product.config` in einem Texteditor. 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.

## Automatisierung

Diese Einstellungen legen die erweiterten Funktionen der Anwendung fest.

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

## Kommunikation Mit Dem Dienst

- **Port für die Kommunikation mit dem Dienst.** Der Automation Manager steuert den Dienst anhand des TCP/IP-Protokolls über den ausgewählten Port. Falls die Nutzung des Standard-Ports auf Ihrem Computer nicht sinnvoll ist, können Sie eine andere Portnummer auswählen. 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. Zu diesem Zeitpunkt werden alte Ereignisse aus der Log-Datenbank gelöscht.
- **Log-Einträge löschen, die älter sind als (Tage).** Gibt an, wie lang Ereignisse in der Log-Datenbank bleiben sollen. Alle Ereignisse, die älter als die angegebene Anzahl von Tagen sind, werden bei jedem Aufräumereignis gelöscht.
- **Protokollnachrichten.** Gibt das Maß an Protokollierung an, das Sie anwenden möchten. Während der Entwicklungs- und Testphasen sollten Sie eine ausführliche Protokollierung aktivieren. Sie sollten alle Nachrichten protokollieren, um die Ausführung von Triggern besser verfolgen zu können. In der Produktionsphase sollten Sie jedoch das Ausmaß der Protokollierung einschränken, sodass nur Fehlermeldungen aufgezeichnet werden.

## Performance

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Enterprise** zur Verfügung.

- **Entfernte 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, 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 vom lokalen Datenträger geladen werden.

Automatisierung Dienst nutzt den folgenden lokalen Ordner, um die entfernten Dateien zwischenspeichern: %PROGRAMDATA%\NiceLabel\NiceLabel 2017\FileCache.

- **Cachedateien aktualisieren (Minuten).** Gibt das Zeitintervall an, nach dem die Dateien im Zwischenspeicher mit den Dateien im ursprünglichen Ordner synchronisiert werden. Dies ist das Zeitlimit, innerhalb dessen das System eventuell Versionen verwenden kann, die nicht aktuell sind.
- **Cachedateien entfernen, die älter sind als (Tage).** Legt das Zeitintervall fest, nach dem alle Dateien aus dem Cache entfernt werden.

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

## Cluster-Unterstützung

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Enterprise** zur Verfügung.

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

### 4.6.6 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 2017 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 2017. Im NiceLabel 2017 [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 2017 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.

**HINWEIS:** Änderungen der Produktebene werden nach Neustart der Anwendung wirksam.

- **Upgrade-Lizenz:** Öffnet den Dialog für Upgrades der Produktebene. Im [NiceLabel 2017 Installationshandbuch](#) finden Sie nähere Angaben zum Lizenz-Upgradeprozess.

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

# 5 Informationen zu Filtern

## 5.1 Informationen Zu Filtern

NiceLabel Automation nutzt Filter, um die Struktur der Daten zu definieren, die von Triggern empfangen werden. Jedes Mal, wenn ein Trigger Daten empfängt, werden diese durch einen oder mehrere Filter geparkt, welche die benötigten Werte extrahieren. Jeder Filter wird anhand von Regeln konfiguriert, die beschreiben, wie Felder in den Daten erkannt werden.

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

### Filtertypen

Weitere Informationen finden Sie in den Themen [Filter für strukturierten Text](#), [Filter für unstrukturierte Daten](#) und [XML-Filter](#).

### Datenstruktur

Die Filterkomplexität hängt von der Datenstruktur ab. Daten, die bereits in strukturierter Form vorliegen, z. B. CSV oder XML, können leicht extrahiert werden. In solchen Fällen sind die Feldnamen bereits den Daten zugeordnet. Die Extraktion von Name:Wert-Paaren geht schnell. Im Fall von unstrukturierten Daten braucht man mehr Zeit für die Definition der Extraktionsregeln. Solche Daten können in Form von exportierten Dokumenten und Berichten aus einem alten System, abgefangener Kommunikation zwischen Geräten, erfassten Druckströmen usw. vorliegen.

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 geparkt werden können. Sie müssen den richtigen Filter für den jeweiligen Eingabedatentyp auswählen. Beispielsweise würden Sie für eingehende CSV-Daten den **Filter für strukturierten Text** 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 Kapitel [Informationen zu Datenstrukturen](#).

### Daten extrahieren

Ein Filter ist nur 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 kann beliebig viele „Datenfilter verwenden“-Aktionen ausführen. Wenn Sie zusammengesetzte Daten erhalten, die nicht von einem einzigen Filter geparkt werden können, können Sie mehrere Filter definieren und ihre jeweiligen Regeln über die Aktion „Datenfilter verwenden“ hintereinander 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“ extrahiert die Werte nicht nur, sondern speichert sie auch in Variablen. Um diesen Prozess zu konfigurieren, müssen Sie die Variablen den jeweiligen Feldern zuordnen. Der Wert des Feldes wird dann in der zugeordneten Variablen gespeichert.

Es empfiehlt sich, Felder und Variablen mit identischen Namen zu definieren. In diesem Fall kann die automatische Zuordnung Variablen mit den gleichnamigen Feldern verbinden, 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 im Thema [Dynamische Struktur aktivieren](#) (zum Filter für strukturierten Text), [Zuweisungsbereiche definieren](#) (zum Filter für unstrukturierte Daten) und [XML-Zuweisungsbereich definieren](#) (zum XML-Filter).

## Aktionen definieren, die für extrahierte Daten ausgeführt werden sollen

Für gewöhnlich möchten Sie bestimmte Aktionen für die 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. In diesem Beispiel würde also nur die fünfte CSV-Zeile gedruckt, die ersten vier Zeilen jedoch nicht. Falls Sie Unterbereiche nutzen, müssen Sie sicherstellen, dass Sie Ihre Aktion unter dem richtigen Platzhalter einbinden.

# 5.2 Filter Für Strukturierten Text Konfigurieren

## 5.2.1 Filter Für Strukturierten Text

Weitere allgemeine Informationen über Filter finden Sie im Abschnitt [Informationen zu Filtern](#).

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

- **Felder werden durch ein Zeichen getrennt.** Häufige Trennzeichen sind Komma oder Semikolon. CSV (durch Kommas getrennte Werte) ist ein typisches Beispiel für eine strukturierte Textdatei.



- **Felder enthalten eine feste Anzahl von Zeichen.** Anders gesagt: Felder werden durch eine feste Spaltenbreite definiert.

Beispiele für strukturierte Textdaten finden Sie im Abschnitt [Textdatenbank](#).

### Struktur definieren

Um die Struktur der Textdatei zu definieren, haben Sie folgende Optionen.

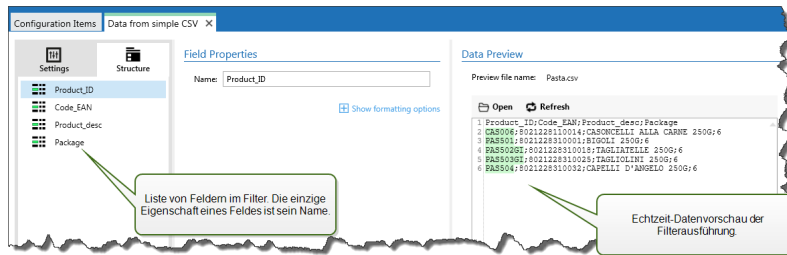
- **Import der Struktur anhand des Textdatei-Assistenten.** Klicken Sie in diesem Fall auf die Schaltfläche **Datenstruktur importieren** in der Menüleiste und folgen Sie den Anweisungen auf dem Bildschirm. Nach Beenden des Assistenten sind die Art der Textdatenbank sowie alle Felder definiert. Falls die erste Datenzeile die Feldnamen enthält, kann der Assistent sie importieren. Falls der Trigger ausschließlich Daten mit identischer Struktur empfangen wird, ist dies die empfohlene Methode.
- **Manuelle Definition der Felder.** In diesem Fall müssen Sie den Typ der Datenfelder (getrennte Felder oder Felder mit fester Länge) und die Feldnamen manuell definieren. Weitere Informationen finden Sie im Abschnitt [Felder definieren](#).
- **Dynamisches Auslesen der Felder.** Falls der Trigger Daten mit unterschiedlicher Struktur empfängt, etwa neue Feldnamen, und Sie den Filter nicht bei jeder Strukturänderung aktualisieren möchten, empfiehlt sich das dynamische Auslesen. Dabei werden alle Felder in den Daten automatisch ausgelesen, unabhängig davon, ob es neue Felder gibt oder einige der alten Felder nicht vorhanden sind. Nach dem Auslesen werden sie automatisch den gleichnamigen Variablen zugeordnet. Weitere Informationen finden Sie im Kapitel [Dynamische Struktur aktivieren](#).

Der Bereich „Datenvorschau“ vereinfacht die Konfiguration. Die Ergebnisse der definierten Filterregeln werden bei jeder Konfigurationsänderung im Vorschaubereich hervorgehoben. So können Sie sehen, welche Daten durch die einzelnen Regeln extrahiert würden.

## 5.2.2 Felder Definieren

Die Definition von Feldern ist bei strukturierten Textdateien sehr einfach. Es gibt zwei Optionen.

- **Felder werden durch Trennzeichen definiert.** In diesem Fall haben Sie Trennzeichen wie Kommas oder Semikolons zwischen den Feldern. Sie müssen lediglich die Feldnamen in der Reihenfolge definieren, in der sie in den vom Trigger empfangenen Daten auftreten.
- **Felder mit fester Breite.** 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 Vorschau Daten an.

- **Name der Vorschau datei.** Gibt die Datei mit Beispieldaten an, die durch den Filter geparkt werden. Die Vorschau datei wird aus der Filterdefinition kopiert. Wenn Sie den Namen der Vorschau datei ändern, wird der neue Name gespeichert.
- **Öffnen.** Wählt eine andere Datei aus, auf die die Filterregeln angewandt werden sollen.
- **Aktualisiere.** Wendet die Filterregeln erneut auf den Inhalt der Vorschau datei an. Der Abschnitt „Datenvorschau“ wird mit dem neuen Ergebnis aktualisiert.

## Formatierungsoptionen

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

- **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 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.

**HINWEIS:** Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. NiceLabel Automation nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im [Knowledge Base-Artikel KB250](#).

- **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 einer 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. NiceLabel Automation 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 NiceLabel-Syntax in tatsächliche Binärzeichen.

**BEISPIEL:** Wenn Sie die Daten „<CR><LF>“ empfangen, fasst NiceLabel Automation 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 neue 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.

## 5.2.3 Dynamische Struktur Aktivieren

Der Filter für strukturierten Text kann die Felder und deren Werte in den Daten automatisch erkennen, wodurch eine manuelle Zuordnung von *Variablen und Feldern* überflüssig wird.

Diese Funktion ist nützlich, wenn der Trigger die Daten der veränderbaren Struktur empfängt. Die Haupt-Datenstruktur ist dieselbe, z. B. Felder werden durch Kommas begrenzt oder identische XML-Struktur, aber **die Reihenfolge**, in der die Felder dargestellt werden, oder **die Anzahl von Feldern** weicht ab; möglicherweise gibt es neue Felder oder alte Felder sind nicht mehr verfügbar. Der Filter erkennt die Struktur automatisch. Gleichzeitig werden die Feldnamen und -werte (**Name:Wert**-Paare) aus den Daten gelesen, was die manuelle Zuordnung von Feldern und Variablen überflüssig macht.

Die Aktion [Datenfilter verwenden](#) zeigt keine Zuordnungsmöglichkeiten an, da die Zuordnung dynamisch erfolgt. Sie müssen nicht einmal die Etikettenvariablen in der Trigger-Konfiguration festlegen. Die Aktion ordnet Feldwerte den gleichnamigen Etikettenvariablen zu, ohne dass die Variablen vom 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.

**HINWEIS:** Wenn ein Feld in den Eingabedaten keine entsprechende Variable auf dem Etikett hat, wird keine Fehlermeldung ausgegeben. Die fehlenden Variablen werden einfach ignoriert.

### 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 die Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden von oben nach unten in der Reihenfolge angewandt, in der sie in der Benutzeroberfläche ausgewählt wurden.

- **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 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.

**HINWEIS:** Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. NiceLabel Automation nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im [Knowledge Base-Artikel KB250](#).

- **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 einer 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. NiceLabel Automation 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 NiceLabel-Syntax in tatsächliche Binärzeichen.

**BEISPIEL:** Wenn Sie die Daten „<CR><LF>“ empfangen, fasst NiceLabel Automation 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 neue 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.

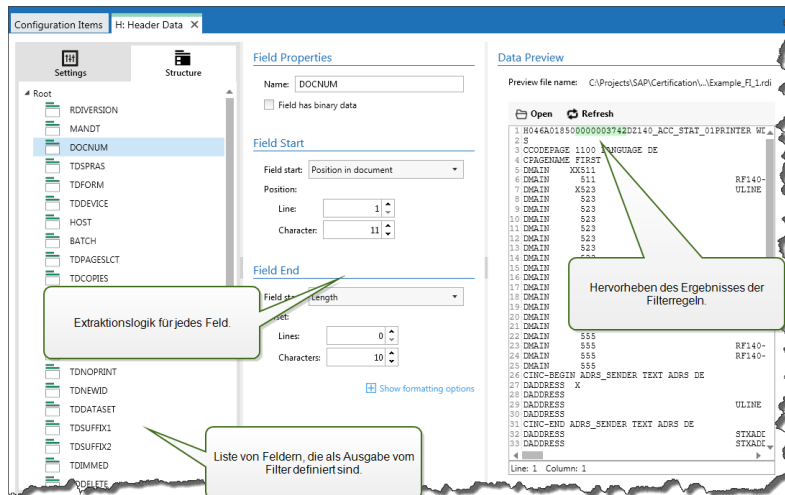
## 5.3 Filter Für Unstrukturierte Daten Konfigurieren

### 5.3.1 Filter Für Unstrukturierte Daten

Weitere allgemeine Informationen über Filter finden Sie im Abschnitt [Informationen zu Filtern](#).

Verwenden Sie diesen Filter, wenn der Trigger unstrukturierte Daten empfängt, z. B. Dokumente und Berichte aus einem alten System, abgefangene Kommunikation zwischen Geräten, erfasste Druckströme usw. 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](#).



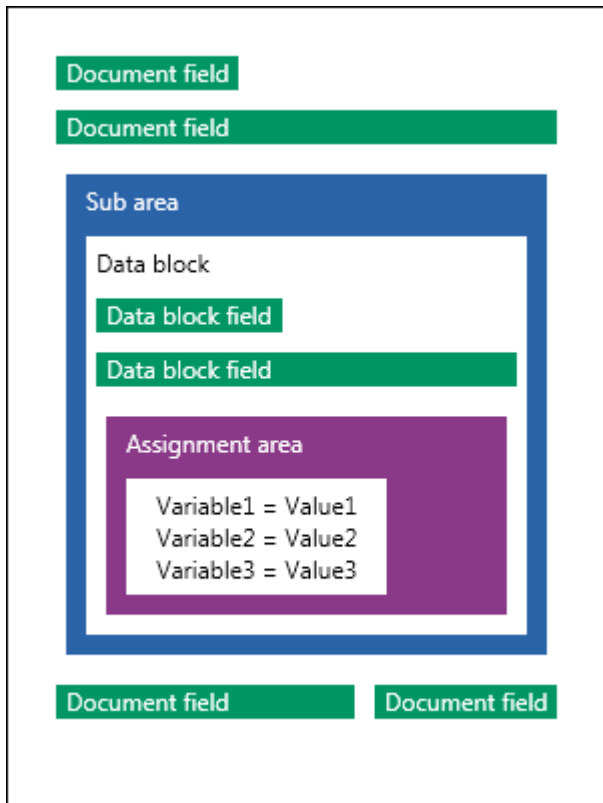
## Struktur definieren

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 wird in der Aktion „Datenfilter verwenden“ eine neue Platzhalterebene definiert, sodass Sie Feldern auf 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. Die 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. Der Zuweisungsbereich kann auf der Stammebene des Dokuments oder im Unterbereich definiert werden. Weitere Informationen finden Sie im Abschnitt [Zuweisungsbereiche definieren](#).

Der Bereich „Datenvorschau“ vereinfacht die Konfiguration. Die Ergebnisse der definierten Filterregeln werden 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 des Datenblocks definiert werden. Die [Name-Wert](#)-Paare können innerhalb des Zuweisungsbereichs definiert werden.



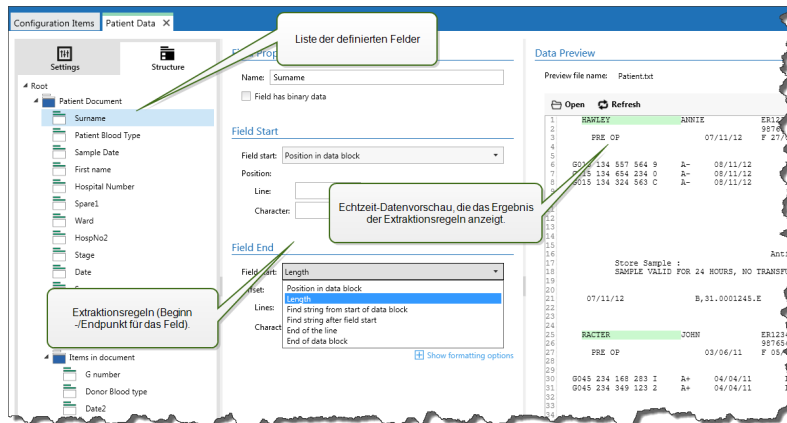
## Allgemein

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

- **Name.** Gibt den Filternamen an. Verwenden Sie einen aussagekräftigen Namen, an dem Sie den Zweck des Filters erkennen können. Sie können ihn jederzeit ändern.
- **Beschreibung.** Bietet die Möglichkeit, die Funktion dieses Filters zu beschreiben. Sie können hier eine kurze Erklärung zum Zweck des Filters eingeben.
- **Codieren.** Gibt die Datencodierung an, mit der dieser Filter arbeiten wird.
- **Leere Zeilen in Datenblöcken ignorieren.** Legt fest, dass kein Fehler gemeldet wird, wenn der Filter leere Feldwerte aus den Datenblöcken extrahiert.

## 5.3.2 Felder Definieren

Wenn Sie ein Feld definieren, müssen Sie seinen Namen sowie eine Regel für die Extraktion des Feldwertes 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.** Der Start-/Endpunkt wird durch die fest codierte Position in den Daten vorgegeben. 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.** Der Start-/Endpunkt wird durch die Position der gesuchten Zeichenfolge definiert. Das erste Zeichen nach der gesuchten Zeichenfolge bestimmt 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 einen solchen Versatz zum Beispiel fest, um die gesuchte Zeichenfolge in die extrahierten Daten einzuschließen.

## Feldende

- **Position im Dokument.** Der Start-/Endpunkt wird durch die fest codierte Position in den



Daten vorgegeben. 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.** Der Start-/Endpunkt wird durch die Position der gesuchten Zeichenfolge definiert. Das erste Zeichen nach der gesuchten Zeichenfolge bestimmt 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 einen solchen Versatz zum Beispiel fest, um die gesuchte Zeichenfolge in die extrahierten Daten einzuschließen.

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

### Formatierungsoptionen

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

- **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 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.

**HINWEIS:** Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. NiceLabel Automation nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im [Knowledge Base-Artikel KB250](#).

- **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 einer 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. NiceLabel Automation 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 NiceLabel-Syntax in tatsächliche Binärzeichen.

**BEISPIEL:** Wenn Sie die Daten „`<CR><LF>`“ empfangen, fasst NiceLabel Automation 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 neue Option aktivieren.

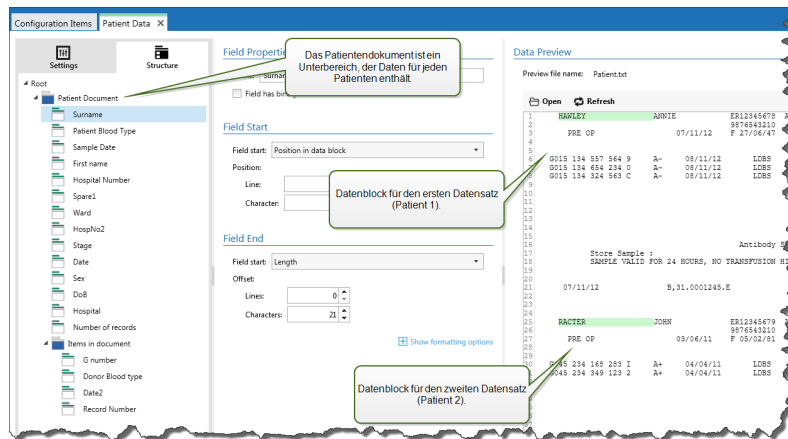
- **Suchen und Löschen von allem vor.** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab Beginn der Daten bis zu der Zeichenfolge. Auch die gefundene Zeichenfolge kann gelöscht werden.
- **Suchen und Löschen von allem nach.** Sucht die angegebene Zeichenfolge und löscht alle Zeichen ab der Zeichenfolge bis zum Ende der Daten. Auch die gefundene Zeichenfolge kann gelöscht werden.

### 5.3.3 Unterbereiche Definieren

Ein Unterbereich ist ein Datenabschnitt, innerhalb dessen es mehrere Datenblöcke gibt, 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 zwei angegebenen Zeichenfolgen gehören zu einem separaten Datenblock. Der Inhalt zwischen der letzten Zeichenfolge und dem Ende der Daten stellt den letzten Datenblock dar.
  - **Block endet mit einer Zeichenfolge.** Gibt an, dass Datenblöcke mit der eingegebenen Zeichenfolge enden. Alle Inhalte zwischen zwei 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 eingegebene Zeichenfolge getrennt werden. Alle Inhalte zwischen zwei angegebenen Zeichenfolgen gehören zu einem separaten Datenblock.

- **Anfang des ersten Datenblocks.** Gibt die Startposition des ersten Datenblocks und damit die Startposition des Unterbereichs an. Für gewöhnlich ist die Startposition 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 und damit die Endposition des Unterbereichs an. 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

Die Felder innerhalb von Unterbereichen werden anhand derselben Parameter konfiguriert wie die auf Stammebene definierten Felder. Weitere Informationen finden Sie im Abschnitt [Felder definieren](#).

**HINWEIS:** 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 mit Beispieldaten an, die durch den Filter geparkt werden. 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.
- **Aktualisiere.** Wendet die Filterregeln erneut auf den Inhalt der Vorschaudatei an. Der Abschnitt „Datenvorschau“ wird mit dem neuen Ergebnis aktualisiert.

## 5.3.4 Zuweisungsbereiche Definieren

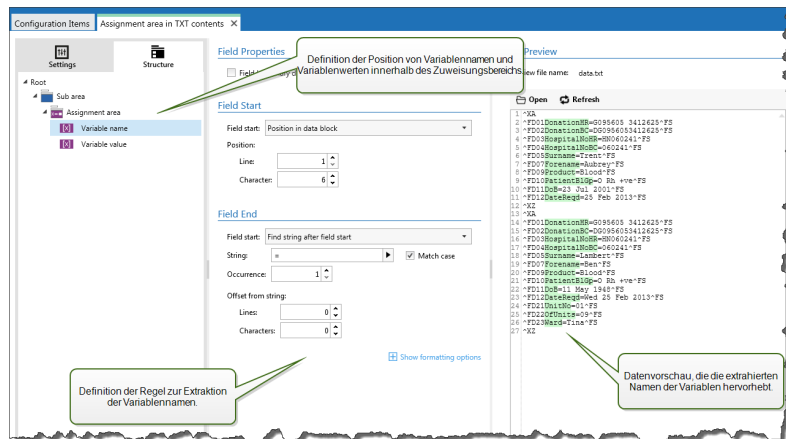
Der Filter für unstrukturierte Daten kann die Felder und deren Werte in den Daten automatisch erkennen, wodurch eine manuelle Zuordnung von *Variablen und Feldern* überflüssig wird.

Diese Funktion ist nützlich, wenn der Trigger die Daten der veränderbaren Struktur empfängt. Die Haupt-Datenstruktur ist dieselbe, z. B. Felder werden durch Kommas begrenzt oder identische XML-Struktur, aber **die Reihenfolge**, in der die Felder dargestellt werden, oder **die Anzahl von Feldern** weicht ab; möglicherweise gibt es neue Felder oder alte Felder sind nicht mehr verfügbar. Der Filter erkennt die Struktur automatisch. Gleichzeitig werden die Feldnamen und -werte (`Name:Wert`-Paare) aus den Daten gelesen, was die manuelle Zuordnung von Feldern und Variablen überflüssig macht.

Die Aktion [Datenfilter verwenden](#) zeigt keine Zuordnungsmöglichkeiten an, da die Zuordnung dynamisch erfolgt. Sie müssen nicht einmal die Etikettenvariablen in der Trigger-Konfiguration festlegen. Die Aktion ordnet Feldwerte den gleichnamigen Etikettenvariablen zu, ohne dass die Variablen vom 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.

**HINWEIS:** Wenn ein Feld in den Eingabedaten keine entsprechende Variable auf dem Etikett hat, wird keine Fehlermeldung ausgegeben. Die fehlenden Variablen werden einfach ignoriert.



## 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

Beim Erstellen des Zuweisungsbereichs definiert der Filter automatisch zwei Platzhalter, welche wiederum das **Name:Wert**-Paar definieren.

- **Variablenname.** Gibt das Feld an, dessen Inhalt den Variablennamen darstellt (**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 den Variablenwert darstellt (**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 im Zuweisungsbereich stellt ein **Name:Wert**-Paar dar. Der Name ist als der Wert zwischen dem 6. Zeichen und dem Gleichheitszeichen in jeder Zeile definiert. Der Wert ist als der Wert zwischen dem Gleichheitszeichen und dem Zeilenende (mit negativem Versatz um drei Zeichen) definiert.

```
^XA
^FD01SpendeHR=G095605 3412625^FS
^FD02SpendeBC=DG0956053412625^FS
^FD03HospitalNoHR=HN060241^FS
^FD04HospitalNoBC=060241^FS
^FD05Nachname=Hartig^FS
^FD07Vorname=Anne^FS
^FD09Produkt=Blut^FS
^FD10PatientBIGp=O Rh +ve^FS
^FD11Geb=27 Juni 1947^FS
^FD12Fälligkeit=25. Dez 2012^FS
^XZ
```

Weitere Informationen finden Sie im Abschnitt [Beispiele](#).

## 5.4 XML-Filter Konfigurieren

### 5.4.1 XML-Filter

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

Weitere allgemeine Informationen über Filter finden Sie im Abschnitt [Informationen zu Filtern](#).

Verwenden Sie diesen Filter, wenn der 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 es sich, die Struktur aus der vorhandenen XML-Beispieldatei zu importieren. Klicken Sie auf **Datenstruktur importieren** in der Menüleiste. Wenn Sie die XML-Struktur importierten, zeigt der Abschnitt „Datenvorschau“ den XML-Inhalt an und hebt 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, müssen Sie ihre Nutzung wie folgt konfigurieren:

- **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 dieses Element nicht obligatorisch ist. Dies entspricht dem Attribut `minOccurs=0` im XML-Schema (XSD-Datei). Die einem solchen Feld zugeordnete Variable hat einen leeren Wert, wenn das Element nicht im XML erscheint.

- **Datenblock.** Gibt an, dass das ausgewählte Element mehrmals vorkommt und Daten für ein einzelnes Etikett bereitstellen wird. Ein Datenblock kann als sich wiederholender Bereich, als Zuweisungsbereich oder beides definiert werden.
  - **Sich wiederholender 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. In der Aktion [Datenfilter verwenden](#) müssen Sie die definierten Felder den jeweiligen Variablen zuordnen. Weitere Informationen finden Sie im Abschnitt [Sich wiederholende Elemente definieren](#).
  - **Zuweisungsbereich.** Gibt an, dass der Datenblock `Name-Wert`-Paare enthält. Die 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. Die Ergebnisse der definierten Filterregeln werden 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.

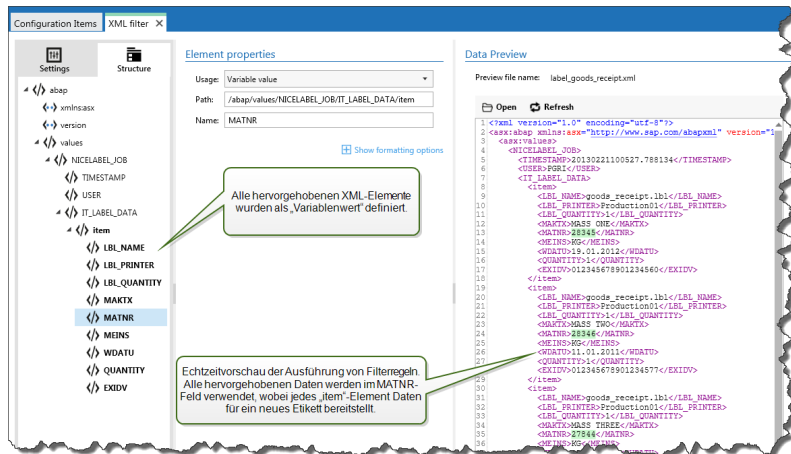
## 5.4.2 XML-Felder Definieren

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

Beim Definieren des XML-Feldes stellen Sie den Wert des ausgewählten Elements als Feld zur Verfügung. Die Filterdefinition stellt dieses Feld in der Aktion „Datenfilter verwenden“ für die Zuordnung zu einer Variablen bereit. Sie können den Wert des Elements oder den Wert des Attributs extrahieren.

So definieren Sie den 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 in der Strukturliste wird in Fettdruck dargestellt, um anzuzeigen, dass es in Verwendung ist.
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 die Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden von oben nach unten in der Reihenfolge angewandt, in der sie in der Benutzeroberfläche ausgewählt wurden.

- **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 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.

**HINWEIS:** Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. NiceLabel Automation nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im [Knowledge Base-Artikel KB250](#).

- **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 einer 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. NiceLabel Automation 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 NiceLabel-Syntax in tatsächliche Binärzeichen.

**BEISPIEL:** Wenn Sie die Daten „<CR><LF>“ empfangen, fasst NiceLabel Automation 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 neue 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.

### 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 mit Beispieldaten an, die durch den Filter geparkt werden. 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.
- **Aktualisiere.** Wendet die Filterregeln erneut auf den Inhalt der Vorschaudatei an. Der Abschnitt „Datenvorschau“ wird mit dem neuen Ergebnis aktualisiert.

## 5.4.3 Sich Wiederholende Elemente Definieren

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

Wenn Sie ein XML-Element haben, das in den XML-Daten mehrmals vorkommt, handelt es sich dabei 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, müssen Sie das Element als **Datenblock definieren** und die Option **Wiederholbares Element** aktivieren. 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 würde `<item>` als Unterebene in der Aktion **Datenfilter verwenden** definiert. Sie müssen die Aktionen „Etikett öffnen“ und „Etikett drucken“ unter diesem Unterebenen-Platzhalter einbetten, damit sie für jede Instanz des Elements `<item>` wiederholt werden (in unserem Beispiel also dreimal).

```
<?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>

<item>
<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>
</item>

<item>
<LBL_NAME>goods_receipt.nlbl</LBL_NAME>
<LBL_PRINTER>Production01</LBL_PRINTER>
<LBL_QUANTITY>1</LBL_QUANTITY>
<MAKTX>MASS TWO</MAKTX>
<MATNR>28346</MATNR>
<MEINS>KG</MEINS>
<WDATU>11.01.2011</WDATU>
<QUANTITY>1</QUANTITY>
<EXIDV>012345678901234577</EXIDV>
</item>

<item>
<LBL_NAME>goods_receipt.nlbl</LBL_NAME>
<LBL_PRINTER>Production01</LBL_PRINTER>
<LBL_QUANTITY>1</LBL_QUANTITY>
<MAKTX>MASS THREE</MAKTX>
<MATNR>27844</MATNR>
<MEINS>KG</MEINS>
<WDATU>07.03.2009</WDATU>
<QUANTITY>1</QUANTITY>
<EXIDV>012345678901234584</EXIDV>
</item>

</IT_LABEL_DATA>
</NICELABEL_JOB>
</asx:values>
</asx:abap>
```

## 5.4.4 XML-Zuweisungsbereich Definieren

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

Der XML-Filter kann die Felder und deren Werte in den Daten automatisch erkennen, wodurch eine manuelle Zuordnung von *Variablen und Feldern* überflüssig wird.

Diese Funktion ist nützlich, wenn der Trigger die Daten der veränderbaren Struktur empfängt. Die Haupt-Datenstruktur ist dieselbe, z. B. Felder werden durch Kommas begrenzt oder identische XML-Struktur, aber **die Reihenfolge**, in der die Felder dargestellt werden, oder **die Anzahl von Feldern** weicht ab; möglicherweise gibt es neue Felder oder alte Felder sind nicht mehr verfügbar. Der Filter erkennt die Struktur automatisch. Gleichzeitig werden die Feldnamen und -werte (**Name:Wert**-Paare) aus den Daten gelesen, was die manuelle Zuordnung von Feldern und Variablen überflüssig macht.

Die Aktion **Datenfilter verwenden** zeigt keine Zuordnungsmöglichkeiten an, da die Zuordnung dynamisch erfolgt. Sie müssen nicht einmal die Etikettenvariablen in der Trigger-Konfiguration festlegen. Die Aktion ordnet Feldwerte den gleichnamigen Etikettenvariablen zu, ohne dass die Variablen vom 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.

**HINWEIS:** Wenn ein Feld in den Eingabedaten keine entsprechende Variable auf dem Etikett hat, wird keine Fehlermeldung ausgegeben. Die fehlenden Variablen werden einfach ignoriert.

Das Element „item“ ist als Datenblock definiert.

Das Element „item“ ist außerdem als Zuweisungsbereich definiert, in dem die Positionen von Variablenamen und -werten definiert sind.

Echtzeitvorschau der Filterregeln. Die Werte der Variablen sind hervorgehoben. Die Namen der Variablen werden durch die XML-Elementennamen vorgegeben.

### 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. Die Etikettenvariable muss denselben Namen haben, damit die automatische Zuordnung funktioniert.
- **Variablenwert.** Gibt das Element an, das den Feldwert enthält. Der Name kann nach Elementname, ausgewähltem Attributwert oder Elementwert definiert werden.

**WARNUNG:** Das XML-Element, das `Name:Wert`-Paare enthält, darf nicht das Stammelement sein, 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 die Funktionen zur Änderung von Zeichenfolgen, die auf ausgewählte Variablen oder Felder angewandt werden. Sie können eine oder mehrere Funktionen auswählen. Die Funktionen werden von oben nach unten in der Reihenfolge angewandt, in der sie in der Benutzeroberfläche ausgewählt wurden.

- **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 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.

**HINWEIS:** Es werden verschiedene Implementierungen der regulären Ausdrücke verwendet. NiceLabel Automation nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im [Knowledge Base-Artikel KB250](#).

- **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 einer 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. NiceLabel Automation 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 NiceLabel-Syntax in tatsächliche Binärzeichen.

**BEISPIEL:** Wenn Sie die Daten „<CR><LF>“ empfangen, fasst NiceLabel Automation 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 neue 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.

### 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 im Abschnitt [Beispiele](#).

## 5.5 Etiketten- Und Druckernamen Anhand Von Eingabedaten Einstellen

Normalerweise werden Filter verwendet, um Werte aus empfangenen Daten zu extrahieren und diese Werte zwecks Druck an die Etikettenvariablen zu senden. In diesem Fall 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 die *Metadaten* bereitstellen; dabei handelt es sich um Werte, die zwar bei der Verarbeitung durch NiceLabel Automation verwendet, aber nicht auf das Etikett gedruckt werden, z. B. Etikettenname, Druckernamen, Etikettenmenge oder andere Informationen.

So nutzen Sie die Werte von Metadatenfeldern im Druckprozess.

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.** Sie müssen die Aktion **Etikett öffnen** neu konfigurieren, 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 im Abschnitt [Beispiele](#).

# 6 Trigger konfigurieren

## 6.1 Informationen Zu Triggern

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht nicht in jedem NiceLabel Automation-Produkt zur Verfügung.

NiceLabel Automation ist eine ereignisbasierte Anwendung, die Aktionen bei Änderungen in Bezug auf die überwachten Ereignisse 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. Die Hauptaufgabe des Triggers besteht darin, die Änderung im Ereignis zu erkennen, vom Ereignis bereitgestellte Daten zu empfangen und dann Aktionen auszuführen. Die meisten Trigger warten passiv darauf, dass das jeweilige Ereignis eintritt, aber es gibt zwei Ausnahmen. Der **Datenbank-Trigger** ist ein aktiver Trigger, der die überwachte Datenbank regelmäßig auf Änderungen überprüft. Der **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 die Etikettenobjekte enthalten. Bevor diese Werte genutzt werden können, müssen sie jedoch aus den empfangenen Daten extrahiert und in Variablen gespeichert werden. Die Filter definieren die 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“.

Tritt das Ereignis ein, werden die von ihm bereitgestellten Eingabedaten in der temporären Datei im `%temp%`-Ordner des jeweiligen Dienstbenutzers gespeichert. Die interne Variable `DataFileName` referenziert den Speicherort der temporären Datei. Nachdem der Trigger die Ausführung abgeschlossen hat, wird die Datei gelöscht.

### Trigger-Eigenschaften

Um den Trigger zu konfigurieren, müssen Sie die Methode des Datenempfangs und die auszuführenden Aktionen definieren. Optional können Sie auch Variablen verwenden. Es gibt drei Bereiche in der Trigger-Konfiguration.

- **Einstellungen.** Definiert die Hauptparameter des ausgewählten Triggers. Sie können das Ereignis definieren, das der Trigger auf Veränderungen überwachen soll, oder den Kanal für eingehende Kommunikation festlegen. Die Einstellungen umfassen auch die Auswahl der Skriptprogrammierungs-Engine sowie Sicherheitseinstellungen. Die verfügbaren Optionen hängen von der Art des Triggers ab. Weitere Informationen finden Sie im

Abschnitt [Trigger-Typen](#).

- **Variablen.** In diesem Abschnitt 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 im Abschnitt [Variablen](#).
- **Aktionen.** In diesem Abschnitt 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 geparkt und in Aktionen verwendet werden.
- **Trigger für serielle Schnittstelle.** Überwacht die eingehende Kommunikation an der seriellen Schnittstelle RS232. Die Inhalte des Eingangstroms können durch Filter geparkt und in Aktionen verwendet werden. Außerdem können die Daten auch in vorgegebenen Zeitintervallen vom externen Gerät abgerufen werden.
- **Datenbank-Trigger.** Überwacht die Datensatz-Änderungen in den SQL-Datenbanktabellen. Die Inhalte des ausgegebenen Daten-Sets können geparkt 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.
- **TCP/IP Server Trigger.** Überwacht den eingehenden Rohdatenstrom, der am definierten Socket ankommt. Die Inhalte des Eingangstroms können durch Filter geparkt und in Aktionen verwendet werden. Kann bidirektional mit Feedback erfolgen.
- **HTTP Server Trigger.** Überwacht den eingehenden Datenstrom im HTTP-Format, der am definierten Socket ankommt. Die Inhalte des Eingangstroms können durch Filter geparkt und in Aktionen verwendet werden. Benutzerauthentifizierung kann aktiviert werden. Ist bidirektional, stellt Feedback bereit.
- **Webdienst-Trigger.** Überwacht den eingehenden Datenstrom, der bei der definierten Webdienstmethode ankommt. Die Inhalte des Eingangstroms können durch Filter geparkt und in Aktionen verwendet werden. Ist bidirektional, stellt Feedback bereit.

### Fehlerhandhabung in Triggern

- **Konfigurationsfehler.** Wenn der Trigger nicht korrekt oder nicht vollständig konfiguriert wurde, weist er einen Fehlerstatus auf. 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 Fehler liegt in einer tieferen Ebene der



Konfiguration, breitet sich aber bis in die höheren Ebenen aus, weswegen die Fehlerposition leicht zu finden ist.

**BEISPIEL:** Falls Sie eine Aktion mit Fehlerstatus haben, zeigen alle Aktionen auf den höheren Ebenen die Fehlersituation an und das Fehlersymbol wird auf der Registerkarte „Aktionen“ und im Triggernamen angezeigt.

- **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

Siehe Abschnitt [Feedback zum Status von Druckaufträgen](#).

## 6.2 Trigger Definieren

### 6.2.1 Dateitrigger

Weitere allgemeine Informationen über Trigger erhalten Sie im Abschnitt [Informationen zu Triggern](#).

Das Dateitrigger-Ereignis tritt ein, wenn eine überwachte Datei oder eine Reihe von Dateien in einem überwachten Ordner geändert wird. Die Ablage einer neuen Datei löst ebenfalls einen Trigger aus. Je nach Trigger-Konfiguration informiert das Windows-System den Trigger über die geänderten Dateien, oder der Trigger selbst verwaltet eine Liste mit Zeitstempeln zu Dateiänderungen und löst aus, wenn die Datei einen neueren Zeitstempel aufweist.

Typische Nutzung: Das vorhandene Geschäftssystem führt eine Transaktion durch, die wiederum eine Trigger-Datei im gemeinsam genutzten Ordner erzeugt. Der Inhalt der Daten 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

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.** Bietet die Möglichkeit, die Funktion dieses Triggers zu beschreiben. Sie können hier eine kurze Erklärung zum Zweck des Triggers eingeben.

- **Angegebene Datei finden.** Gibt den Pfad und Dateinamen der Datei an, die auf Änderungen überwacht werden soll.
- **Einen Satz von Dateien im angegebenen Ordner finden.** Gibt den Pfad zu dem Ordner, der auf Dateiänderungen überwacht werden soll, sowie die Dateinamen an. Sie können Standard-Windows-Platzhalter wie „\*“ und „?“ verwenden. Einige Dateitypen sind in der Dropdown-Liste vordefiniert, aber Sie können auch eigene Typen angeben.

**HINWEIS:** Wenn Sie den 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.** Die Anwendung reagiert 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.** Die Anwendung überprüft den Ordner im angegebenen Intervall 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 zugreifen soll.

- **Datei exklusiv öffnen.** Gibt an, dass die Triggerdatei im exklusiven Modus geöffnet werden soll. Auf diese Weise kann keine andere Anwendung gleichzeitig auf die Datei zugreifen. Dies ist die Standardauswahl.
- **Datei nur mit Leserecht öffnen.** Gibt an, dass die Triggerdatei im Nur-Lesen-Modus geöffnet werden soll.
- **Datei mit Lese- und Schreibrechten öffnen.** Gibt an, dass die Triggerdatei im Lesen/Schreiben-Modus geöffnet werden soll.
- **Intervall, in dem versucht wird, die Datei erneut zu öffnen.** Gibt das Zeitintervall an, in dem NiceLabel Automation versuchen wird, 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 Zeitstempels, sondern auch anhand der Dateilänge. Die Änderungen am Zeitstempel der Datei werden eventuell nicht erkannt; daher löst diese Option die jeweiligen Aktionen auch bei einer Änderung der Dateigröße aus.

- **Leere Triggerdateien ignorieren.** Falls die Triggerdatei keinen Inhalt hat, wird sie ignoriert. Die Aktionen werden nicht ausgeführt.
- **Triggerdatei löschen** Nach Erkennung der Änderung in der Triggerdatei und Auslösung des Triggers wird die Datei gelöscht. Durch Aktivierung dieser Option können bereits verarbeitete Dateien aus dem Ordner entfernt werden.

**HINWEIS:** NiceLabel Automation erstellt immer ein Backup der empfangenen Triggerdaten (in diesem Fall des Inhalts der Triggerdatei) und speichert es unter einem eindeutigen Dateinamen. 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 keine alten Daten drucken.
- **Änderungen nachverfolgen wenn Trigger inaktiv ist.** Gibt an, ob der Trigger für Änderungen ausgelöst werden soll, die eintreten, während der Trigger inaktiv ist. Wenn 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, können die vorhandenen Triggerdateien verarbeitet werden.

### Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikation mit der Druck-Engine fest.

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

- **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 die Variable passen, oder ob fehlende Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet NiceLabel Automation Fehler und unterbricht den Druckprozess, wenn Sie versuchen, einen zu langen Wert in einer Etikettenvariablen zu speichern oder den Wert für eine nicht vorhandene Etikettenvariable anzugeben.

- **Übermäßig lange Variableninhalte ignorieren.** Datenwerte, welche die Länge der Variablen laut Definition im Etiketten-Designer überschreiten, werden gekürzt, um in die Variable zu passen. 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 10 Zeichen. Wenn diese Option aktiviert ist, werden Werte mit mehr als 10 Zeichen auf die ersten 10 Zeichen gekürzt; alle Zeichen nach dem zehnten werden also ignoriert.

- **Fehlende Etikettenvariablen ignorieren.** Wenn Sie den Druck mit [Befehlsdateien](#) (z. B. JOB-Dateien) ausführen, ignoriert der Druckprozess alle Variablen, die in der Befehlsdatei festgelegt (anhand des [SET](#)-Befehls), aber nicht im Etikett definiert sind. Wenn Sie versuchen, den Wert für eine nicht vorhandene Etikettenvariable festzulegen, wird kein Fehler ausgegeben. Eine ähnliche Verarbeitung wird ausgeführt, wenn Sie Zuweisungsbereiche im Filter definieren, um alle *Name:Wert*-Paare zu extrahieren, aber weniger Variablen im Etikett definiert haben.

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

- **Scripting-Sprache.** Legt die für den Trigger aktivierte Scripting-Sprache fest. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden dieselbe 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 den 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 Fehler bei der Ausführung der Aktion aufgetreten sind. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu erhalten, die das Problem hervorgerufen haben, sodass eine spätere Problemlösung möglich ist.

**WARNUNG:** Aktivieren Sie die Unterstützung für überwachtes Drucken; andernfalls kann NiceLabel Automation den Fehler während der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt [Synchroner Druckmodus](#).

**HINWEIS:** NiceLabel Automation speichert die empfangenen Daten in einer temporären Datei, die direkt nach Abschluss der Trigger-Ausführung gelöscht wird. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie im Abschnitt [Interne Variablen](#).

## Sicherheit

- **Trigger sperren und verschlüsseln.** Aktiviert den Trigger-Schutz. Wenn er aktiviert ist, wird der Trigger gesperrt und kann nicht bearbeitet werden; außerdem werden Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

## 6.2.2 Trigger Für Serielle Schnittstelle

Weitere allgemeine Informationen über Trigger erhalten Sie im 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](#).

### 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.** Bietet die Möglichkeit, die Funktion dieses Triggers zu beschreiben. Sie können hier eine kurze Erklärung zum Zweck des Triggers eingeben.
- **Schnittstelle.** Gibt die Nummer der seriellen Schnittstelle (COM) an, wo eingehende 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 entsprechen müssen, die dem Gerät an der seriellen Schnittstelle zugeordnet wurden.

- **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 \(Steuercodes\)](#).

- **Datenpolling verwenden.** Gibt an, dass der Trigger das Gerät aktiv auf Daten abfragen soll. Der Trigger sendet die im Feld [Eingabe von Sonderzeichen \(Steuercodes\)](#) angegebenen Befehle zu den vorgegebenen Zeitintervallen. Dieses Feld kann binäre Zeichen enthalten. Weitere Informationen finden Sie im Abschnitt [Eingabe von Sonderzeichen \(Steuercodes\)](#).

## Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikation mit der Druck-Engine fest.

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

- **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 die Variable passen, oder ob fehlende Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet NiceLabel Automation Fehler und unterbricht den Druckprozess, wenn Sie versuchen, einen zu langen Wert in einer Etikettenvariablen zu speichern oder den Wert für eine nicht vorhandene Etikettenvariable anzugeben.

- **Übermäßig lange Variableninhalte ignorieren.** Datenwerte, welche die Länge der Variablen laut Definition im Etiketten-Designer überschreiten, werden gekürzt, um in die Variable zu passen. 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 10 Zeichen. Wenn diese Option aktiviert ist, werden Werte mit mehr als 10 Zeichen auf die ersten 10 Zeichen gekürzt; alle Zeichen nach dem zehnten werden also ignoriert.

- **Fehlende Etikettenvariablen ignorieren.** Wenn Sie den Druck mit [Befehlsdateien](#) (z. B. JOB-Dateien) ausführen, ignoriert der Druckprozess alle Variablen, die in der Befehlsdatei festgelegt (anhand des [SET](#)-Befehls), aber nicht im Etikett definiert sind. Wenn Sie versuchen, den Wert für eine nicht vorhandene Etikettenvariable festzulegen, wird kein Fehler ausgegeben. Eine ähnliche Verarbeitung wird ausgeführt, wenn Sie Zuweisungsbereiche im Filter definieren, um alle *Name:Wert*-Paare zu extrahieren, aber weniger Variablen im Etikett definiert haben.

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

- **Scripting-Sprache.** Legt die für den Trigger aktivierte Scripting-Sprache fest. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden dieselbe 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 den 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 Fehler bei der Ausführung der Aktion aufgetreten sind. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu erhalten, die das Problem hervorgerufen haben, sodass eine spätere Problemlösung möglich ist.

**WARNUNG:** Aktivieren Sie die Unterstützung für überwachtes Drucken; andernfalls kann NiceLabel Automation den Fehler während der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt [Synchroner Druckmodus](#).

**HINWEIS:** NiceLabel Automation speichert die empfangenen Daten in einer temporären Datei, die direkt nach Abschluss der Trigger-Ausführung gelöscht wird. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie im Abschnitt [Interne Variablen](#).

### Sicherheit

- **Trigger sperren und verschlüsseln.** Aktiviert den Trigger-Schutz. Wenn er aktiviert ist, wird der Trigger gesperrt und kann nicht bearbeitet werden; außerdem werden Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

## 6.2.3 Datenbank-Trigger

Weitere allgemeine Informationen über Trigger erhalten Sie im Abschnitt [Informationen zu Triggern](#).

Das 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 die Daten aus der Datenbank in den vorgegebenen Zeitintervallen ab.

Typische Nutzung: Das vorhandene 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 die 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.** Bietet die Möglichkeit, die Funktion dieses Triggers zu beschreiben. Sie können hier eine kurze Erklärung zum Zweck des Triggers eingeben.
- **Datenbankverbindung.** Gibt die Zeichenfolge für die Verbindung zur Datenbank an. Klicken Sie auf die Schaltfläche **Definieren**, um ein Datenbank-Dialogfeld zu öffnen, in dem Sie eine Verbindung zur Datenbank konfigurieren können, einschließlich Datenbanktyp, Tabellennamen und Zugangsdaten für Benutzer. Sie müssen eine Verbindung zu der Datenbank herstellen, die einen Zugriff über SQL-Befehle ermöglicht. Daher können Sie den Datenbank-Trigger nicht für die automatische Erkennung von Daten in CSV-Textdateien (mit durch Kommas getrennten Werten) und Microsoft Excel Tabellenkalkulationen verwenden.

**HINWEIS:** 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 Verbindungen mit Datenbanken finden Sie im Abschnitt [Zugriff auf Datenbanken](#).

- **Datenbank in folgenden Zeitintervallen prüfen.** Gibt das Zeitintervall an, in dem die Datenbank auf Datensätze geprüft wird.
- **Erkennungsoptionen und Erweitert.** Diese Optionen ermöglichen Ihnen eine Feinabstimmung des Erkennungsmechanismus für Datensätze. Wenn die Datensätze von der Datenbank empfangen werden, zeigt die Registerkarte „Aktion“ automatisch das Objekt „Für jeden Datensatz“ an, wo Sie die Tabellenfelder Etikettenvariablen zuordnen können.

#### **Alle auf dem einzigartigen Feldwert basierenden Datensätze übernehmen**

In diesem Fall überwacht der Trigger ein festgelegtes autoinkrementelles numerisches 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 (`Tabellennamen`), 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.

**HINWEIS:** 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

In diesem Fall werden alle Datenbänke aus der Tabelle abgerufen und danach aus der Tabelle gelöscht. Um diese Option zu konfigurieren, müssen Sie den Namen der Tabelle auswählen, in der sich die Datensätze befinden (**Tabellenname**) und den primären Schlüssel in der Tabelle (**Schlüsselfelder**) angeben. Obwohl eine Tabelle 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 im aktuellen Trigger empfangenen Datensätze auf einmal gelöscht. 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 aller weiteren 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

**HINWEIS:**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 den primären Index enthält.

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

Das Feld **ID** 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 die Nutzung der Variablen vorgegeben.

### Beispiel einer SQL-Aktualisierungsanweisung, wenn 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

In diesem Fall werden alle Datenbänke aus der Tabelle abgerufen und danach 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 den aktuellen Wert des Felds zu referenzieren (**Aktualisierungswert**).

Obwohl eine Tabelle 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 aller weiteren 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

**HINWEIS:** 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 den primären Index enthält.

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

Das Feld **ID** 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 die Nutzung der Variablen vorgegeben. Das Feld **UpdateValue** wird in der Trigger-Konfiguration im Feld **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 liegen die SQL-Anweisungen für die Datensatzextraktion und die Feldaktualisierungen ganz bei Ihnen. 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 Werte aus Tabellenfeldern 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.

So rufen Sie die Datensätze ab, die noch nicht gedruckt wurden. 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 aufgeführten Beispieltabelle werden zwei Datensätze mit den ID-Werten 2 und 3 extrahiert. Der erste Datensatz wurde bereits gedruckt und wird ignoriert.

### Beispiel für eine 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 Variablenamen in Ihrer SQL-Anweisung setzen, damit die Variable als solche erkannt werden kann. Sie können für die 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 andere Felder im Datensatz als `:ProductID` oder `:CodeEAN` referenzieren, oder auch Variablen referenzieren, 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 Ergebnisse anzuzeigen, klicken Sie auf die Schaltfläche „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 die entsprechenden Testwerte für sie angeben.

**HINWEIS:** Falls Sie die Datenvorschau geöffnet und soeben einige Variablen zum Skript hinzugefügt haben, klicken Sie zweimal auf die Schaltfläche **Test** (um den Abschnitt „Datenvorschau“ zu schließen und zu öffnen), um die Liste mit den Variablen in der Vorschau zu aktualisieren.

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

**HINWEIS:** Falls nötig, können Sie die Datensätze auch aktualisieren, während die Aktionen noch ausgeführt werden. Weitere Informationen finden Sie im Abschnitt Variablen.

## Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikation mit der Druck-Engine fest.

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

- **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 die Variable passen, oder ob fehlende Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet NiceLabel Automation Fehler und unterbricht den Druckprozess, wenn Sie versuchen, einen zu langen Wert in einer Etikettenvariablen zu speichern oder den Wert für eine nicht vorhandene Etikettenvariable anzugeben.

- **Übermäßig lange Variableninhalte ignorieren.** Datenwerte, welche die Länge der Variablen laut Definition im Etiketten-Designer überschreiten, werden gekürzt, um in die Variable zu passen. 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 10 Zeichen. Wenn diese Option aktiviert ist, werden Werte mit mehr als 10 Zeichen auf die ersten 10 Zeichen gekürzt; alle Zeichen nach dem zehnten werden also ignoriert.

- **Fehlende Etikettenvariablen ignorieren.** Wenn Sie den Druck mit [Befehlsdateien](#) (z. B. JOB-Dateien) ausführen, ignoriert der Druckprozess alle Variablen, die in der Befehlsdatei festgelegt (anhand des [SET](#)-Befehls), aber nicht im Etikett definiert sind.

Wenn Sie versuchen, den Wert für eine nicht vorhandene Etikettenvariable festzulegen, wird kein Fehler ausgegeben. Eine ähnliche Verarbeitung wird ausgeführt, wenn Sie Zuweisungsbereiche im Filter definieren, um alle *Name:Wert*-Paare zu extrahieren, aber weniger Variablen im Etikett definiert haben.

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

- **Scripting-Sprache.** Legt die für den Trigger aktivierte Scripting-Sprache fest. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden dieselbe 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 den 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 Fehler bei der Ausführung der Aktion aufgetreten sind. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu erhalten, die das Problem hervorgerufen haben, sodass eine spätere Problemlösung möglich ist.

**WARNUNG:** Aktivieren Sie die Unterstützung für überwachtetes Drucken; andernfalls kann NiceLabel Automation den Fehler während der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt [Synchroner Druckmodus](#).

**HINWEIS:** NiceLabel Automation speichert die empfangenen Daten in einer temporären Datei, die direkt nach Abschluss der Trigger-Ausführung gelöscht wird. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie im Abschnitt [Interne Variablen](#).

## Sicherheit

- **Trigger sperren und verschlüsseln.** Aktiviert den Trigger-Schutz. Wenn er aktiviert ist, wird der Trigger gesperrt und kann nicht bearbeitet werden; außerdem werden Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

## 6.2.4 TCP/IP Server Trigger

Weitere allgemeine Informationen über Trigger erhalten Sie im Abschnitt [Informationen zu Triggern](#).

Das TCP/IP-Triggerereignis tritt ein, wenn Daten am überwachten Socket (IP-Adresse und Portnummer) empfangen werden.

Typische Nutzung: Das vorhandene Geschäftssystem führt eine Transaktion durch, wodurch wiederum Daten über ein bestimmtes Socket an den NiceLabel Automation Server gesendet werden. Der Inhalt der Daten 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

**HINWEIS:** 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.** Bietet die Möglichkeit, die Funktion dieses Triggers zu beschreiben. Sie können hier eine kurze Erklärung zum Zweck des Triggers eingeben.
- **Schnittstelle.** Gibt die Nummer der Schnittstelle an, wo eingehende 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 Sicherheitsfragen finden Sie unter [Zugriff auf Ihre Trigger sichern](#).

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

- **Maximale Anzahl gleichzeitiger 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.

**HINWEIS:** 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.** Gibt an, 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.** Gibt an, dass der Trigger immer auslöst, wenn die erforderliche Zeichenfolge 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 Zeichenfolge, welche das Trigger-Ereignis bestimmt, wird nicht aus den Daten entfernt, sondern mit ihnen übermittelt. Der Trigger wird den empfangenen Datenstrom vervollständigen.
- **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 die 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 diese Antwort erwartet, sobald die Ausführung der Aktionen abgeschlossen ist. 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 Kommunikation mit der Druck-Engine fest.

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.



- **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 die Variable passen, oder ob fehlende Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet NiceLabel Automation Fehler und unterbricht den Druckprozess, wenn Sie versuchen, einen zu langen Wert in einer Etikettenvariablen zu speichern oder den Wert für eine nicht vorhandene Etikettenvariable anzugeben.

- **Übermäßig lange Variableninhalte ignorieren.** Datenwerte, welche die Länge der Variablen laut Definition im Etiketten-Designer überschreiten, werden gekürzt, um in die Variable zu passen. 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 10 Zeichen. Wenn diese Option aktiviert ist, werden Werte mit mehr als 10 Zeichen auf die ersten 10 Zeichen gekürzt; alle Zeichen nach dem zehnten werden also ignoriert.

- **Fehlende Etikettenvariablen ignorieren.** Wenn Sie den Druck mit [Befehlsdateien](#) (z. B. JOB-Dateien) ausführen, ignoriert der Druckprozess alle Variablen, die in der Befehlsdatei festgelegt (anhand des [SET](#)-Befehls), aber nicht im Etikett definiert sind. Wenn Sie versuchen, den Wert für eine nicht vorhandene Etikettenvariable festzulegen, wird kein Fehler ausgegeben. Eine ähnliche Verarbeitung wird ausgeführt, wenn Sie Zuweisungsbereiche im Filter definieren, um alle *Name:Wert*-Paare zu extrahieren, aber weniger Variablen im Etikett definiert haben.

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

- **Scripting-Sprache.** Legt die für den Trigger aktivierte Scripting-Sprache fest. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden dieselbe 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 den 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 Fehler bei der Ausführung der Aktion aufgetreten sind. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu erhalten, die das Problem hervorgerufen haben, sodass eine spätere Problemlösung möglich ist.

**WARNUNG:** Aktivieren Sie die Unterstützung für überwachtes Drucken; andernfalls kann NiceLabel Automation den Fehler während der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt [Synchroner Druckmodus](#).

**HINWEIS:** NiceLabel Automation speichert die empfangenen Daten in einer temporären Datei, die direkt nach Abschluss der Trigger-Ausführung gelöscht wird. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie im Abschnitt [Interne Variablen](#).

### Sicherheit

- **Trigger sperren und verschlüsseln.** Aktiviert den Trigger-Schutz. Wenn er aktiviert ist, wird der Trigger gesperrt und kann nicht bearbeitet werden; außerdem werden Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

## 6.2.5 HTTP Server Trigger

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

Weitere allgemeine Informationen über Trigger erhalten Sie im Abschnitt [Informationen zu Triggern](#).

Das HTTP-Triggerereignis tritt ein, 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 müssen den Standard-HTTP-Header enthalten. Die Drittanwendung muss die POST- oder GET-Anfrage verwenden und Daten im Nachrichtentext oder in der Abfragefolge bereitstellen. Es spielt keine Rolle, welchen Internet-Medientyp (MIME-Typ oder Content-Typ) sie im Nachrichtentext verwenden. NiceLabel Automation empfängt die Nachricht, und Sie können einen Filter zur Extraktion der erforderlichen Daten aus dem Nachrichteninhalt definieren.

Typische Nutzung: Das vorhandene Geschäftssystem führt eine Transaktion durch, wodurch wiederum als HTTP POST-Nachricht formatierte Daten über ein bestimmtes Socket an den NiceLabel Automation Server gesendet werden. Der Inhalt der Daten 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 die extrahierten Daten auf Etiketten. Weitere Informationen zum Parsen und Extrahieren von Daten finden Sie unter [Informationen zu Filtern](#).

### Daten bereitstellen

Sie können die Daten für den HTTP-Trigger mit einer der folgenden Methoden bereitstellen. 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.

Ein typischer URL mit Abfragefolge sieht folgendermaßen aus:

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

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

Die Abfragefolge setzt sich für gewöhnlich aus einer Reihe von `Name:Wert`-Paaren zusammen, wobei in jedem Paar der Feldname und der Wert durch ein Gleichheitszeichen (=) getrennt werden. Die Paare in der Reihe werden jeweils durch ein Et-Zeichen (&) getrennt. Eine typische Abfragefolge würde Werte für Felder (Variablen) also folgendermaßen bereitstellen:

```
feld1=wert1&feld2=wert2&feld3=wert3
```

Der HTTP-Trigger verfügt über integrierte Unterstützung für die Extraktion der Werte aller Felder und deren Speicherung in den Variablen mit identischen Namen; daher müssen Sie für die Extraktion von Werten aus der Abfragefolge keine Filter definieren.

- 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, werden ihre jeweiligen Werte eingefügt. Gibt es keine solchen Variablen im Etikett, werden die Werte ignoriert und es wird keine Fehlermeldung angezeigt.
- Sie müssen Variablen im Trigger nur definieren, wenn Sie deren Werte im Rahmen einer Aktion innerhalb des jeweiligen Triggers benötigen. Um alle in der Abfragefolge enthaltenen Werte zu erhalten, müssen Sie einfach die Variablen definieren, deren Namen mit den Feldern 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 der HTTP-Anfrage

Sie müssen die POST-Abfragemethode verwenden, um die Nachricht im Textkörper der HTTP-Anfrage bereitzustellen.

Sie können beliebige Daten und Datenstrukturen im Textkörper verwenden, sofern die NiceLabel Automation-Filter diese Daten verarbeiten können. Der Inhalt kann im XML- oder CSV-Format, als Klartext oder sogar in Form von Binärdaten (Base64-codiert) vorliegen. Denken Sie daran, dass die Daten mit Filtern geparkt 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.** Bietet die Möglichkeit, die Funktion dieses Triggers zu beschreiben. Sie können hier eine kurze Erklärung zum Zweck des Triggers eingeben.

## Kommunikation

**HINWEIS:** Dieser Trigger unterstützt Internet Protocol Version 6 (IPv6).

In diesem Bereich können Sie die obligatorische Portnummer sowie verschiedene Feedback-Optionen konfigurieren. Sie können die Standard-HTTP-Antwortcodes verwenden, um den Erfolg der ausgeführten Aktion anzuzeigen. In komplexeren Fällen können Sie auch benutzerdefinierte Inhalte an die datengebende Anwendung senden; dabei kann es sich um eine einfache Feedback-Zeichenfolge oder Binärdaten wie eine Etikettenvorschau oder einen Druckstrom handeln.

Der typische URL zur Verbindung mit dem 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.
- **Schnittstelle.** Gibt die Nummer der Schnittstelle an, wo eingehende 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 Sicherheitsfragen finden Sie unter [Zugriff auf Ihre Trigger sichern](#).

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

- **Pfad.** Gibt den optionalen Pfad im 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 einen anderen URL ausgelöst werden können. Wenn Sie nicht wissen, welchen Pfad Sie verwenden sollen, behalten Sie die Standardeinstellung (/) bei.

**TIPP:** Diese Option ist in NiceLabel Automation Enterprise verfügbar.

- **Sichere Verbindung (HTTPS).** Aktiviert die sichere Transportschicht für Ihre HTTP-Nachrichten und verhindert Abhörangriffe. Weitere Informationen zur Einrichtung von HTTPS finden Sie unter [Sicheres Übertragungsprotokoll \(HTTPS\) nutzen](#).
- **Abfragefolge.** Gibt die Name-Wert-Paare im URL an. Dieser Parameter ist optional, da die Daten normalerweise im Textkörper der HTTP-Anfrage bereitgestellt werden.

- **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, der anzeigt, dass die Daten erfolgreich empfangen wurden; dies jedoch sagt nichts über den Erfolg der Trigger-Aktionen aus.

Diese Option gibt vor, dass der Trigger die Antwort nicht direkt nach Empfang der Daten sendet, sondern wartet, bis alle Aktionen ausgeführt wurden und danach einen Antwort-Code sendet, der die erfolgreiche Ausführung anzeigt. Wenn diese Option aktiviert ist, können Sie die 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.

**HINWEIS:**Falls Sie Feedback zum Druckprozess senden möchten, müssen Sie den **synchronen** Druckmodus aktivieren. Weitere Informationen finden Sie unter [Synchroner Druckmodus](#).

- **Maximale Anzahl gleichzeitiger Anfragen.** Gibt die maximale Anzahl gleichzeitiger eingehender Verbindungen an. Dies definiert die Anzahl von Clients, die gleichzeitig Daten an den Trigger senden können. Diese Zahl hängt auch von der Hardwareleistung Ihres Servers ab.
- **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 einfach manuell ein. Die Antwortdaten werden im angegebenen Medientyp formatiert und als Feedback gesendet. **Variable** aktiviert den Variablen-Medientyp. Wählen Sie bei aktivierter Option eine Variable aus (oder erstellen Sie eine Variable), die den Medientyp enthält.

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

- **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.

**HINWEIS:**Wenn Ihre Ausgabe ausschließlich aus binären Inhalten besteht (z. B.

```
application/octet-streamapplication/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 unter [Zusammengesetzte Werte verwenden](#).

## Authentifizierung

- **Keine.** Es wird keine Authentifizierungsmethode verwendet.
- **Benutzer.** Gibt an, dass eingehende Nachrichten einen Benutzernamen und ein Passwort enthalten. Der Trigger akzeptiert nur HTTP-Nachrichten mit übereinstimmenden Zugangsdaten. Weitere Informationen zu Sicherheitsfragen finden Sie unter [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. Der Trigger akzeptiert nur HTTP-Nachrichten mit entsprechenden Anmeldedaten für NiceLabel Control Center Benutzer, die eine bestimmten Anwendungsgruppe angehören.
  - **Gruppe.** Im NiceLabel Control Center können mehrere Anwendungsgruppen 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.

**HINWEIS:** 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 werden anhand ihrer Anmeldedaten authentifiziert, 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 Kommunikation mit der Druck-Engine fest.

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

- **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 die Variable passen, oder ob fehlende Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet NiceLabel Automation Fehler und unterbricht den Druckprozess, wenn Sie versuchen, einen zu langen Wert in einer Etikettenvariablen zu speichern oder den Wert für eine nicht vorhandene Etikettenvariable anzugeben.

- **Übermäßig lange Variableninhalte ignorieren.** Datenwerte, welche die Länge der Variablen laut Definition im Etiketten-Designer überschreiten, werden gekürzt, um in die Variable zu passen. 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 10 Zeichen. Wenn diese Option aktiviert ist, werden Werte mit mehr als 10 Zeichen auf die ersten 10 Zeichen gekürzt; alle Zeichen nach dem zehnten werden also ignoriert.

- **Fehlende Etikettenvariablen ignorieren.** Wenn Sie den Druck mit [Befehlsdateien](#) (z. B. JOB-Dateien) ausführen, ignoriert der Druckprozess alle Variablen, die in der Befehlsdatei festgelegt (anhand des [SET](#)-Befehls), aber nicht im Etikett definiert sind. Wenn Sie versuchen, den Wert für eine nicht vorhandene Etikettenvariable festzulegen, wird kein Fehler ausgegeben. Eine ähnliche Verarbeitung wird ausgeführt, wenn Sie Zuweisungsbereiche im Filter definieren, um alle *Name:Wert*-Paare zu extrahieren, aber weniger Variablen im Etikett definiert haben.

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

- **Scripting-Sprache.** Legt die für den Trigger aktivierte Scripting-Sprache fest. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden dieselbe 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 den 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 Fehler bei der Ausführung der Aktion aufgetreten sind. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu erhalten, die das Problem hervorgerufen haben, sodass eine spätere Problemlösung möglich ist.

**WARNUNG:** Aktivieren Sie die Unterstützung für überwacht Drucken; andernfalls kann NiceLabel Automation den Fehler während der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt [Synchroner Druckmodus](#).

**HINWEIS:** NiceLabel Automation speichert die empfangenen Daten in einer temporären Datei, die direkt nach Abschluss der Trigger-Ausführung gelöscht wird. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie im Abschnitt [Interne Variablen](#).

## Sicherheit

- **Trigger sperren und verschlüsseln.** Aktiviert den Trigger-Schutz. Wenn er aktiviert ist, wird der Trigger gesperrt und kann nicht bearbeitet werden; außerdem werden Aktionen verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

## 6.2.6 Webdienst-Trigger

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Enterprise** zur Verfügung.

Weitere allgemeine Informationen über Trigger erhalten Sie im Abschnitt [Informationen zu Triggern](#).

Das Webdienst-Triggerereignis tritt ein, wenn Daten am überwachten Socket (IP-Adresse und Portnummer) empfangen werden. Die Daten müssen der SOAP-Notation entsprechen (als HTTP-Nachricht codierte XML-Daten). Die Webdienst-Schnittstelle ist im WSDL-Dokument beschrieben, das mit jedem definierten Webdienst-Trigger bereitgestellt wird.

Der Webdienst kann ein Feedback zum Druckauftrag bereitstellen, aber Sie müssen dazu den **synchronen** Verarbeitungsmodus aktivieren. Weitere Informationen finden Sie im Abschnitt [Feedback zum Status von Druckaufträgen](#).

Normalerweise wird der Webdienst von Programmierern genutzt, um den Etikettendruck in ihre eigenen Anwendungen einzubinden. Das vorhandene Geschäftssystem führt eine Transaktion durch, wodurch wiederum als SOAP-Nachricht formatierte Daten über ein bestimmtes Socket an den NiceLabel Automation Server gesendet werden. Die Daten können in einem strukturierten Format wie CSV und XML oder aber in einem Legacy-Format bereitgestellt werden. 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

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.** Bietet die Möglichkeit, die Funktion dieses Triggers zu beschreiben. Sie können hier eine kurze Erklärung zum Zweck des Triggers eingeben.

## Kommunikation

**HINWEIS:** Dieser Trigger unterstützt Internet Protocol Version 6 (IPv6).

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

- **Schnittstelle.** Gibt die Nummer der Schnittstelle an, wo eingehende 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 Sicherheitsfragen finden Sie unter [Zugriff auf Ihre Trigger sichern](#).

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

- **Sichere Verbindung (HTTPS).** Aktiviert die sichere Transportschicht für Ihre HTTP-Nachrichten und verhindert Abhörangriffe. Weitere Informationen zur Einrichtung von HTTPS finden Sie unter [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 beim Ausführen definierter Aktionen. Die Ausführung von Aktionen kann überwacht werden. Der Trigger meldet den Erfolgsstatus für jedes Ausführungsereignis. Um Statusmeldungen aus dem Druckprozess zu aktivieren, müssen

Sie den [Synchroner Druckmodus](#) aktivieren.

Folgende Methoden (Funktionen) werden vom Webdienst-Trigger angeboten:

- **ExecuteTrigger.** Diese Methode nimmt Daten zur Verarbeitung entgegen und gibt ein optionales Statusfeedback aus. Einer der Eingabeparameter aktiviert bzw. deaktiviert das 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 und enthält immer die Fehler-ID und die Fehlerbeschreibung.
- **ExecuteTriggerWithResponse.** Diese Methode nimmt Daten zur Verarbeitung entgegen und gibt das benutzerdefinierte Statusfeedback aus. Die Webdienst-Antwort ist konfigurierbar. Sie können beliebige Daten in jeder beliebigen Struktur zurücksenden. Sie können binäre Daten in der Antwort verwenden.
- **ExecuteTriggerAndSetVariables.** Ähnlich wie die obige Methode **ExecuteTrigger**, bietet aber zusätzliche eingehende Parameter, welche die formatierte Liste mit *Name-Wert*-Paaren entgegennehmen. Der Trigger parst die Liste automatisch, extrahiert Werte und speichert sie in den gleichnamigen Variablen, sodass Sie keinen eigenen Filter für die Datenextraktion definieren müssen.
- **ExecuteTriggerAndSetVariablesWithResponse.** Ähnlich wie die obige Methode **ExecuteTriggerWithResponse**, bietet aber zusätzliche eingehende Parameter, welche die formatierte Liste mit *Name-Wert*-Paaren entgegennehmen. Der Trigger parst die Liste automatisch, extrahiert Werte und speichert sie in den gleichnamigen Variablen, sodass Sie keinen eigenen Filter für die Datenextraktion definieren müssen.

Weitere Informationen über die Struktur der Nachrichten, die Sie an die eine oder die andere Methode senden können, finden Sie im folgenden Abschnitt, [WSDL](#).

## WSDL

Gibt den Stil der SOAP-Nachrichten an. Dies kann entweder ein **Remoteprozeduraufruf-Stil (RPC)** oder ein **Dokument-Stil** sein. Wählen Sie den Stil aus, der von der Anwendung unterstützt wird, die Daten für NiceLabel Automation bereitstellt.

Das WSDL-Dokument (Web Service Description Language) definiert die Eingabe- und Ausgabeparameter des Webdienstes.

Wenn Sie den Webdienst-Trigger an Port 12345 definieren, ihn in Automation Manager implementieren und dann starten, steht das WSDL unter folgender Adresse zur Verfügung:

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

Das WSDL bietet verschiedene Methoden an, die allesamt Daten für den Trigger bereitstellen. Sie müssen diejenige auswählen, die für Ihre Zwecke am besten geeignet ist.

- Die Methoden mit *WithResponse* im Namen ermöglichen es Ihnen, angepasste Antworten zu senden, etwa benutzerdefinierte Fehlermeldungen, eine Etikettenvorschau, PDF-Dateien, Druckdateien (\*.PRN) und ähnliches. Auch die Methoden ohne

*WithResponse* im Namen geben Feedback aus, aber Sie können die Antwort nicht anpassen. Das Feedback enthält standardmäßige Fehlermeldungen.

- Die Methoden mit *SetVariables* im Namen ermöglichen es Ihnen, eine Liste mit Variablen in zwei vordefinierten Formaten bereitzustellen; deren Werte werden extrahiert und automatisch den entsprechenden Variablen zugeordnet. 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 ist folgender:

```
<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>
```

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

- **text.** Dies ist die Eingabe-Zeichenfolge, die von dem Filter geparkt werden kann, der in der Konfiguration definiert ist. Normalerweise ist die Zeichenfolge als CSV oder XML strukturiert, damit sie problemlos von einem Filter geparkt werden kann, aber Sie können auch ein anderes Textformat verwenden.
- **wait.** Dies ist ein boolesches Feld, das angibt, ob auf die Druckauftragsstatus-Rückmeldung gewartet werden und ob der Webdienst Feedback ausgeben soll. Für *Wahr* verwenden Sie **1**, für *Falsch* verwenden Sie **0**. Abhängig von dem Methodentyp, den Sie auswählen, gibt es entweder eine vordefinierte Antwort, oder Sie können eine benutzerdefinierte Antwort senden.

Es gibt die folgenden 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**, wenn es keine Probleme bei der Verarbeitung der Daten gab, und eine Ganzzahl größer **0**, wenn Probleme aufgetreten sind. Die Anwendung, welche den Webdienst-Aufruf an NiceLabel Automation ausführt, kann die Antwort als Fehlerindikator verwenden.
- **errorText.** Dieser Zeichenfolgen-Wert enthält die Statusantwort für den Druckauftrag, wenn während der Trigger-Verarbeitung ein Fehler ausgegeben wurde.

**HINWEIS:** Wenn während der Trigger-Verarbeitung ein Fehler aufgetreten ist, 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 der 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 ist folgender:

```
<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>
```

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

- **text.** Dies ist die Eingabe-Zeichenfolge, die von dem Filter geparkt werden kann, der in der Konfiguration definiert ist. Normalerweise ist die Zeichenfolge als CSV oder XML strukturiert, damit sie problemlos von einem Filter geparkt werden kann, aber Sie können auch ein anderes Textformat verwenden.
- **wait.** Dies ist ein boolesches Feld, das angibt, ob auf die Druckauftragsstatus-Rückmeldung gewartet werden und ob der Webdienst Feedback ausgeben soll. Für *Wahr* verwenden Sie **1**, für *Falsch* verwenden Sie **0**. Abhängig von dem Methodentyp, den Sie auswählen, gibt es entweder eine vordefinierte Antwort, oder Sie können eine benutzerdefinierte Antwort senden.

Es gibt die folgenden optionalen Ausgabevariablen (Sie erhalten ihre Werte, wenn Sie sie anfordern, indem Sie **wait** auf **1** setzen:

- **ExecuteTriggerWithResponseResult.** Die als Ganzzahl ausgegebene Antwort enthält den Wert **0**, wenn es keine Probleme bei der Verarbeitung der Daten gab, und eine Ganzzahl größer **0**, wenn Probleme aufgetreten sind. Die Anwendung, welche den Webdienst-Aufruf an NiceLabel Automation ausführt, kann die Antwort als Fehlerindikator verwenden.
- **responseData.** Die benutzerdefinierte Antwort, die Sie bei der Konfiguration des Webdienst-Triggers festlegen können. Die Antwort erfolgt in Form von base64-codierten Daten.

- **errorText.** Dieser Zeichenfolgen-Wert enthält die Statusantwort für den Druckauftrag, wenn während der Trigger-Verarbeitung ein Fehler ausgegeben wurde.

**HINWEIS:** Wenn während der Trigger-Verarbeitung ein Fehler aufgetreten ist, 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.

### ExecuteTriggerAndSetVariables-Methode

Der Hauptteil der Definition ist folgender:

```
<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>
```

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

- **text.** Dies ist die Eingabe-Zeichenfolge, die von dem Filter geparkt werden kann, der in der Konfiguration definiert ist. Normalerweise ist die Zeichenfolge als CSV oder XML strukturiert, damit sie problemlos von einem Filter geparkt werden kann, aber Sie können auch ein anderes Textformat verwenden.
- **wait.** Dies ist ein boolesches Feld, das angibt, ob auf die Druckauftragsstatus-Rückmeldung gewartet werden und ob der Webdienst Feedback ausgeben soll. Für *Wahr* verwenden Sie **1**, für *Falsch* verwenden Sie **0**. Abhängig von dem Methodentyp, den Sie auswählen, 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, wird das jeweilige *Name:Wert*-Paar nicht beachtet. Wenn Sie die Liste von Variablen und ihren Werten in dieser Methode angeben, müssen Sie keine Datenextraktion in den Filtern definieren. Der Trigger wird das gesamte Parsing für Sie übernehmen.

Die Inhalte für „variableData“ können mithilfe einer der zwei verfügbaren Strukturen angegeben werden.

#### XML-Struktur

Die Variablen werden innerhalb des Root-Elements `<Variables />` in der XML-Datei angegeben. Der Variablenname wird anhand des Attributnamens, der Variablenwert anhand des Elementwerts bereitgestellt.

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

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

**HINWEIS:** Sie müssen Ihre XML-Daten innerhalb des CDATA-Abschnitts einbetten. **CDATA**, also **Zeichendaten**, ist ein Abschnitt des Elementinhalts, der dem Parser mitteilt, dass er als normale Zeichendaten interpretiert werden soll, nicht als Markup. Der gesamte Inhalt wird als Zeichendaten verwendet, zum Beispiel wird `<element>ABC</element>` als `&lt;element&gt;ABC&lt;/element&gt;` interpretiert. Ein CDATA-Abschnitt beginnt mit der Folge `<![CDATA[` und endet mit der Folge `]]>`. Schließen Sie Ihre XML-Daten einfach in diese Folgen ein.

### Abgegrenzte Struktur

Die Variablen werden in Form eines Textstroms bereitgestellt. Jedes *Name:Wert*-Paar wird in einer neuen Zeile angegeben. Der Variablenname steht auf der linken Seite des Gleichheitszeichens (=), der Variablenwert auf der rechten Seite.

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

Es gibt die folgenden optionalen Ausgabevariablen (Sie erhalten ihre Werte, wenn Sie sie anfordern, indem Sie **wait** auf 1 setzen:

- **ExecuteTriggerAndSetVariablesResult.** Die als Ganzzahl ausgegebene Antwort enthält den Wert 0, wenn es keine Probleme bei der Verarbeitung der Daten gab, und eine Ganzzahl größer 0, wenn Probleme aufgetreten sind. Die Anwendung, welche den Webdienst-Aufruf an NiceLabel Automation ausführt, kann die Antwort als Fehlerindikator verwenden.
- **errorText.** Dieser Zeichenfolgen-Wert enthält die Statusantwort für den Druckauftrag, wenn während der Trigger-Verarbeitung ein Fehler ausgegeben wurde.

**HINWEIS:** Wenn während der Trigger-Verarbeitung ein Fehler aufgetreten ist, 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.

### ExecuteTriggerAndSetVariablesWithResponse-Methode

Sie würden diese Methode verwenden, wenn der 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 ist folgender:

```
<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.** Dies ist die Eingabe-Zeichenfolge, die von dem Filter geparkt werden kann, der in der Konfiguration definiert ist. Normalerweise ist die Zeichenfolge als CSV oder XML strukturiert, damit sie problemlos von einem Filter geparkt werden kann, aber Sie können auch ein anderes Textformat verwenden.
- **wait.** Dies ist ein boolesches Feld, das angibt, ob auf die Druckauftragsstatus-Rückmeldung gewartet werden und ob der Webdienst Feedback ausgeben soll. Für *Wahr* verwenden Sie **1**, für *Falsch* verwenden Sie **0**. Abhängig von dem Methodentyp, den Sie auswählen, 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, wird das jeweilige *Name:Wert*-Paar nicht beachtet. Wenn Sie die Liste von Variablen und ihren Werten in dieser Methode angeben, müssen Sie keine Datenextraktion in den Filtern definieren. Der Trigger wird das gesamte Parsing für Sie übernehmen.

Die Inhalte für „variableData“ können mithilfe einer der zwei verfügbaren Strukturen angegeben werden.

### XML-Struktur

Die Variablen werden innerhalb des Root-Elements `<Variables />` in der XML-Datei angegeben. Der Variablenname wird anhand des Attributnamens, der Variablenwert anhand des Elementwerts bereitgestellt.

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

**HINWEIS:** Sie müssen Ihre XML-Daten innerhalb des CDATA-Abschnitts einbetten.

]]>]]>. Schließen Sie Ihre XML-Daten einfach in diese Folgen ein.

### Abgegrenzte Struktur

Die Variablen werden in Form eines Textstroms bereitgestellt. Jedes *Name:Wert*-Paar wird in einer neuen Zeile angegeben. Der Variablenname steht auf der linken Seite des Gleichheitszeichens (=), der Variablenwert auf der rechten Seite.

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

Es gibt die folgenden optionalen Ausgabevariablen (Sie erhalten ihre Werte, wenn Sie sie anfordern, indem Sie **wait** auf 1 setzen:

- **ExecuteTriggerAndSetVariablesWithResponseResult.** Die als Ganzzahl ausgegebene Antwort enthält den Wert 0, wenn es keine Probleme bei der Verarbeitung der Daten gab, und eine Ganzzahl größer 0, wenn Probleme aufgetreten sind. Die Anwendung, welche den Webdienst-Aufruf an NiceLabel Automation ausführt, kann die Antwort als Fehlerindikator verwenden.
- **responseData.** Die benutzerdefinierte Antwort, die Sie bei der Konfiguration des Webdienst-Triggers festlegen können. Die Antwort erfolgt in Form von base64-codierten Daten.
- **errorText.** Dieser Zeichenfolgen-Wert enthält die Statusantwort für den Druckauftrag, wenn während der Trigger-Verarbeitung ein Fehler ausgegeben wurde.

**HINWEIS:** Wenn während der Trigger-Verarbeitung ein Fehler aufgetreten ist, 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.

### Sonstiges

Die Optionen im Bereich **Feedback von der Print Engine** legen die Kommunikation mit der Druck-Engine fest.

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

- **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 die Variable passen, oder ob fehlende Etikettenvariablen ignoriert werden sollen. Standardmäßig meldet NiceLabel Automation Fehler und unterbricht den Druckprozess, wenn Sie versuchen, einen zu langen Wert in einer Etikettenvariablen zu speichern oder den Wert für eine nicht vorhandene Etikettenvariable anzugeben.



- **Übermäßig lange Variableninhalte ignorieren.** Datenwerte, welche die Länge der Variablen laut Definition im Etiketten-Designer überschreiten, werden gekürzt, um in die Variable zu passen. 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 10 Zeichen. Wenn diese Option aktiviert ist, werden Werte mit mehr als 10 Zeichen auf die ersten 10 Zeichen gekürzt; alle Zeichen nach dem zehnten werden also ignoriert.

- **Fehlende Etikettenvariablen ignorieren.** Wenn Sie den Druck mit [Befehlsdateien](#) (z. B. JOB-Dateien) ausführen, ignoriert der Druckprozess alle Variablen, die in der Befehlsdatei festgelegt (anhand des [SET](#)-Befehls), aber nicht im Etikett definiert sind. Wenn Sie versuchen, den Wert für eine nicht vorhandene Etikettenvariable festzulegen, wird kein Fehler ausgegeben. Eine ähnliche Verarbeitung wird ausgeführt, wenn Sie Zuweisungsbereiche im Filter definieren, um alle *Name:Wert*-Paare zu extrahieren, aber weniger Variablen im Etikett definiert haben.

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

- **Scripting-Sprache.** Legt die für den Trigger aktivierte Scripting-Sprache fest. Alle **Script ausführen**-Aktionen, die Sie innerhalb eines einzelnen Triggers nutzen, verwenden dieselbe 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 den 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 Fehler bei der Ausführung der Aktion aufgetreten sind. Sie könnten diese Option beispielsweise aktivieren, um die Daten zu erhalten, die das Problem hervorgerufen haben, sodass eine spätere Problemlösung möglich ist.

**WARNUNG:** Aktivieren Sie die Unterstützung für überwachtes Drucken; andernfalls kann NiceLabel Automation den Fehler während der Ausführung nicht erkennen. Weitere Informationen finden Sie im Abschnitt [Synchroner Druckmodus](#).

**HINWEIS:** NiceLabel Automation speichert die empfangenen Daten in einer temporären Datei, die direkt nach Abschluss der Trigger-Ausführung gelöscht wird. Die interne Variable `DataFileName` verweist auf diesen Dateinamen. Weitere Informationen finden Sie im Abschnitt [Interne Variablen](#).

## Sicherheit

- **Trigger sperren und verschlüsseln.** Aktiviert den Trigger-Schutz. Wenn er aktiviert ist, wird der Trigger gesperrt und kann nicht bearbeitet werden; außerdem werden Aktionen

verschlüsselt. Nur Benutzer mit einem Passwort können den Trigger entsperren und bearbeiten.

## 6.3 Variablen Verwenden

### 6.3.1 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 der Filter Werte aus den vom Trigger empfangenen Datenströmen und sendet diese an Variablen. Weitere Informationen finden Sie im Abschnitt [Informationen zu Filtern](#).

Normalerweise sollen die Werte von Variablen an eine zu druckende Etikettenvorlage gesendet werden. Beim Senden von Variablenwerten an Etiketten wird eine automatisierte Zuordnung genutzt: Der Wert der im Trigger definierten Variablen wird an die gleichnamige Variable gesendet, die im Etikett definiert ist. Sie können Variablen auf drei verschiedene Arten definieren:

- **Variablen aus Etikettendatei importieren.** Für die oben beschriebene automatische Zuordnung empfiehlt es sich, Ihre Variablen jedes Mal vom Etikett zu importieren. Dadurch stellen Sie sicher, dass die Variablennamen identisch sind, und können Zeit sparen. 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.

**HINWEIS:** 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 im Abschnitt [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 der 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 die Variable enthalten darf.
- **Feste Länge.** Legt fest, dass der Wert exakt aus einer definierten Anzahl von Zeichen bestehen muss.

**HINWEIS:** Sie müssen die Länge des Variablenwerts für bestimmte Objekte auf dem Etikett begrenzen, zum Beispiel für EAN-13-Barcodes, die genau 13 Zeichen erfordern.

- **Wert erforderlich.** Legt fest, dass die Variable einen Wert enthalten muss
- **Standardwert.** Legt einen Standardwert fest. Wenn der Variablen kein Wert zugewiesen wird, wird immer der Standardwert genutzt.

## 6.3.2 Zusammengesetzte Werte Verwenden

Einige Objekte in der Trigger-Konfiguration akzeptieren zusammengesetzte Werte. Der Inhalt kann dabei 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. Sie können auf diese Schaltfläche klicken, um entweder eine Variable oder ein Sonderzeichen einzufügen.

- **Feste Werte verwenden.** Sie können einen festen Wert für Variablen eingeben.

```
Dies ist ein fester Wert.
```

- **Feste Werte und Daten aus Variablen verwenden.** Sie können außerdem einen zusammengesetzten Wert definieren, der sich aus Werten von Variablen und festen Werten zusammensetzt. 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](#).  
Im folgenden Fall setzt sich der Inhalt aus drei Variablen und einigen festen Daten zusammen.

```
[variable1] // Dies ist ein fester Wert [variable2][variable3]
```

- **Sonderzeichen verwenden.** Sie können auch Sonderzeichen in der Kombination nutzen.

Sie können entweder manuell eingegeben oder eingefügt werden. Weitere Informationen finden Sie im Abschnitt [Eingabe von Sonderzeichen \(Steuercodes\)](#). Im folgenden Fall wird der Wert von `Variable1` mit einigen festen Daten und dem binären Steuerzeichen „Form Feed“ verbunden.

```
[variable1] Form Feed folgt auf diesen festen Text <FF>
```

### 6.3.3 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. Trigger-interne Variablen sind für den jeweiligen Trigger spezifisch.

Interne Variable	Verfügbar in Trigger	Beschreibung
ActionLastErrorDesc	Alle	Stellt die Beschreibung des zuletzt aufgetretenen Fehlers bereit. Sie können diesen Wert in einem Feedback an das Host-System nutzen, um die Ursache des Fehlers zu bestimmen.
ActionLastErrorID	Alle	Stellt die ID des zuletzt aufgetretenen Fehlers 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. „26.2.2013“.

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. „2013“.
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 die der Trigger geantwortet hat. 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, Web- dienst	Stellt die Schnittstellennummer gemäß Definition im Trigger bereit.
RemoteHttpIp	HTTP	Stellt den Hostnamen des Geräts/Computers bereit, das/der sich mit dem Trigger verbindet.
Remotelp	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.

### 6.3.4 Globale Variablen

Globale Variablen sind Variablen, die auf mehreren Etiketten verwendet werden können. Sie werden außerhalb der Etikettendatei definiert und speichern den zuletzt genutzten Wert. Globale Variablen werden für gewöhnlich als globale Zähler definiert. Eine globale Variable stellt 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 in den **Optionen** (Datei>Werkzeuge>Optionen) konfigurieren.

Standardmäßig ist NiceLabel Automation für die Nutzung globaler Variablen vom lokalen Computer konfiguriert. Der Standard-Speicherort lautet:

```
%PROGRAMDATA%\NiceLabel\Global Variables
```

Die globalen 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.

**HINWEIS:**Die Definition und der aktuelle Wert für globale Variablen kann in einer Datei oder im Control Center (für **NiceLabel LMS Enterprise** und **NiceLabel LMS Pro** Produkte) gespeichert werden.

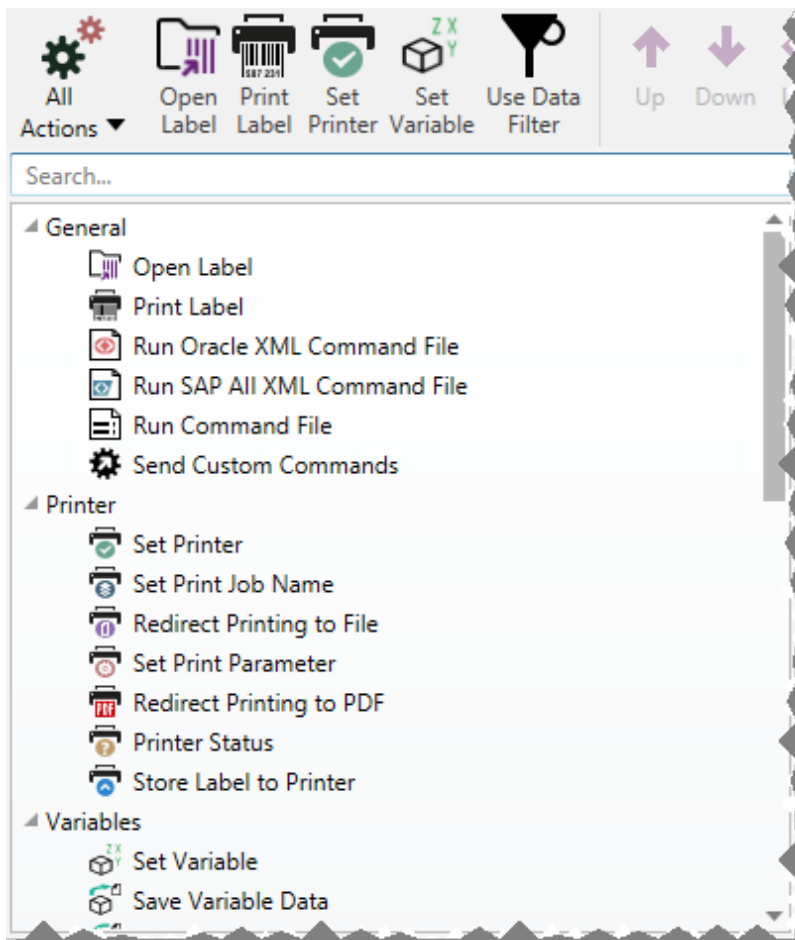
## 6.4 Aktionen Verwenden

### 6.4.1 Aktionen

Der Abschnitt „Aktionen“ gibt die Liste von Aktionen an, die bei jedem Auslösen des Triggers ausgeführt werden.

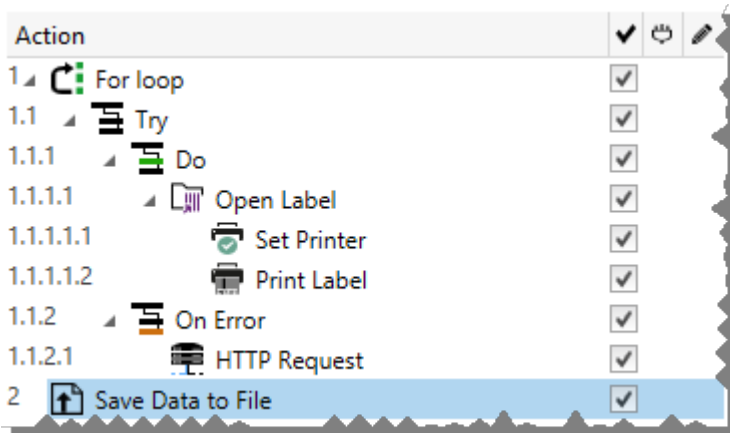
#### 6.4.1.1 Aktionen Definieren

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.



### 6.4.1.2 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.



**HINWEIS:**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.

### 6.4.1.3 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.

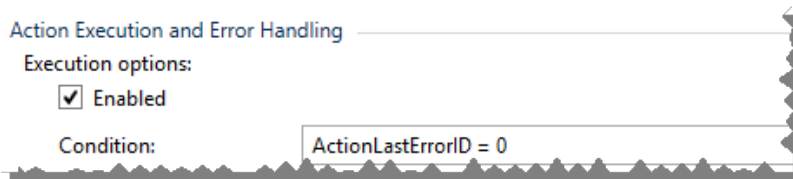
Es gibt zwei Ausnahmen. Die Aktionen **FOR Schleife** und **Datenfilter verwenden** führen geschachtelte Aktionen mehrmals aus: Im Fall von **FOR Schleife** werden Aktionen so häufig ausgeführt, wie in den Aktionseigenschaften festgelegt, und im Fall von **Datenfilter verwenden** einmal für jeden Datensatz in einem vom verbundenen Filter zurückgegebenen Daten-Set.

NiceLabel 2017 wird als Dienst unter einem bestimmten Windows-Benutzerkonto ausgeführt und übernimmt die Sicherheitsberechtigungen des Kontos. Weitere Informationen finden Sie im Kapitel „Im Dienstmodus ausführen“ im NiceLabel Automation Benutzerhandbuch.

### 6.4.1.4 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.

#### 6.4.1.5 Aktionen Mit Konfigurationsfehlern Erkennen

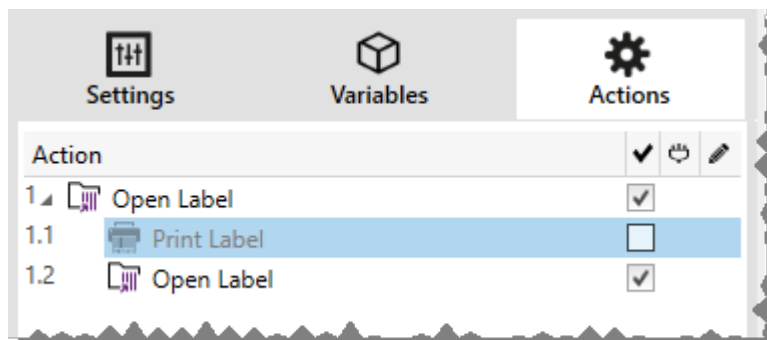
Wenn eine Aktion nicht vollständig konfiguriert ist, wird sie mit einem roten Ausrufezeichen markiert. Solche Aktionen können nicht ausgeführt werden. Sie können solche Aktionen in die **Aktion**-Liste aufnehmen, müssen aber die Konfiguration abschließen, bevor Sie den Trigger starten können. Wenn eine 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.

#### 6.4.1.6 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.

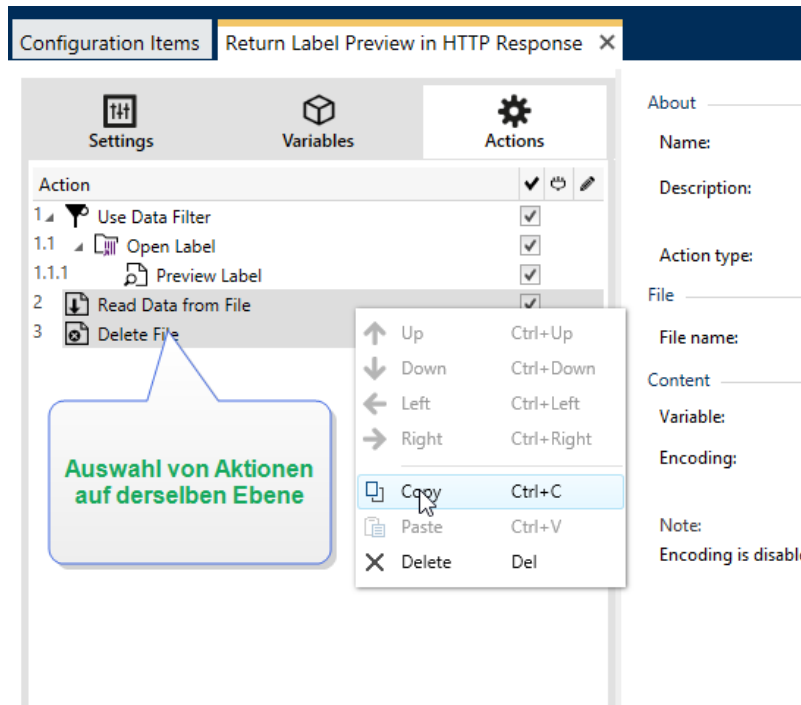
### 6.4.1.7 Aktionen Kopieren

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.

**HINWEIS:** Mehrere Aktionen können nur unter derselben übergeordneten Aktion ausgewählt werden, d. h. alle ausgewählten Aktionen müssen sich auf derselben Ebene befinden.



### 6.4.1.8 In Der Aktionsliste Navigieren

Wählen Sie eine definierte Aktion per Mausclick 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.

### 6.4.1.9 Die Aktionen Beschreiben

Die Gruppe **Über** ermöglicht Ihnen eine Beschreibung aller NiceLabel 2017 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 unterscheiden 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.

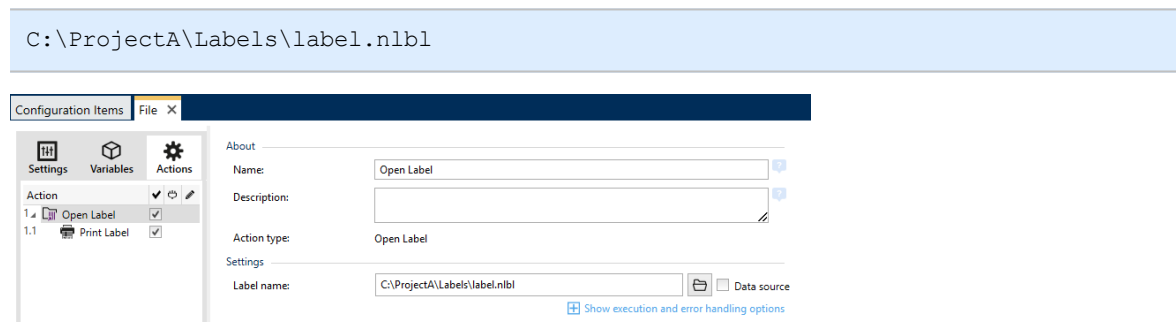
## 6.4.2 Allgemein

### 6.4.2.1 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 2017 das Etikett `label.nlbl` aus dem Ordner `C:\ProjectA\Labels`.



Wenn das angegebene Etikett nicht auffindbar ist, versucht NiceLabel 2017, es an alternativen Speicherorten zu finden. Weitere Informationen finden Sie im Abschnitt **Workflows**.

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 2017 unterstützt die Verwendung relativer Pfade zum Referenzieren Ihrer Etikettendatei. Das Stammverzeichnis ist dabei immer der Ordner, in dem die Konfigurationsdatei (MISX) gespeichert ist.

Mit der folgenden Syntax wird das Etikett relativ zum Speicherort der Konfigurationsdatei geladen. Zuerst wird das Etikett im Ordner `ProjectA` gesucht, der sich zwei Ebenen über dem aktuellen Ordner befindet, und dann in `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.

**HINWEIS:** Verwenden Sie für Netzwerkressourcen die UNC-Syntax.

## Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.2.2 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 des Befehls [Drucker einstellen](#) umgehen.

Um Etikettendruck mit hoher Leistung zu erzielen, aktiviert NiceLabel 2017 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 datengegebende Anwendung senden kann. Weitere Informationen finden Sie im Abschnitt „Synchrone Modus“ im NiceLabel Automation Benutzerhandbuch.

**HINWEIS:** 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.

**HINWEIS:** 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.

**HINWEIS:** 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. Der NiceLabel 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 abrufen. 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 Mengeninformatoren. 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.

**TIPP:** Alle **Erweiterten** Gruppenwerte können entweder fest codiert oder durch eine vorhandene oder neu hinzugefügte Variable dynamisch bereitgestellt werden.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen *ActionLastErrorId* und *ActionLastErrorDesc* gespeichert.

#### 6.4.2.3 Oracle XML-Befehlsdatei Ausführen

**DESIGNER-PRODUKTEBENE INFO**Die beschriebene Funktion ist Teil des Moduls Automation Builder und ist in **NiceLabel oracoroa** und **NiceLabel LMS Pro** verfügbar.

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.



## 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. In **Actions** tab, add the action to execute the command file, such as [Oracle XML-Befehlsdatei ausführen](#), Run Oracle XML Command File, or Run SAP All XML Command File (last two actions are available in Automation Builder).

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.

## Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.2.4 SAP All XML-Befehlsdatei Ausführen

**DESIGNER-PRODUKTEBENE INFO**Die beschriebenen Funktionen sind Teil des Moduls Automation Builder und in **NiceLabel LMS Enterprise** und **NiceLabel LMS Pro** verfügbar.

Diese Aktion führt den Druck mit Daten aus einer Datei im SAP All XML-Format aus.

NiceLabel Automation bietet interne Unterstützung für XML-Dateien mit der „SAP All XML“-Struktur, welche in der SAP-Software definiert werden.

Nutzen Sie diese Aktion als einen Kurzbefehl. Sie hilft Ihnen bei der Ausführung von SAP All 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 All XML-Spezifikationen entsprechen. Weitere Informationen finden Sie im Abschnitt „SAP All 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.

#### **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. In **Actions** tab, add the action to execute the command file, such as [SAP All XML-Befehlsdatei ausführen](#), Run Oracle XML Command File, or Run SAP All XML Command File (last two actions are available in Automation Builder).

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 All XML-Befehlsdatei.

- **Dateiname:** ausgewählte SAP All 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.

- **Etikettename:** 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.2.5 Befehlsdatei Ausführen

Diese Aktion führt Befehle aus, die in einer ausgewählten Befehlsdatei enthalten sind. Alle **Dateityp**-Optionen stellen Befehle bereit, die NiceLabel 2017 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.

### 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. In **Actions** tab, add the action to execute the command file, such as [Befehlsdatei ausführen](#), Run Oracle XML Command File, or Run SAP All XML Command File (last two actions are available in Automation Builder).

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.2.6 Benutzerdefinierte Befehle Senden

Diese Aktion führt die eingegebenen benutzerdefinierten NiceLabel Befehle aus.

Schachteln Sie diese Aktion unter der Aktion [Etikett öffnen](#). Dies ermöglicht eine Referenzierung des Etiketts, auf das die Befehle angewandt werden. Weitere Informationen finden Sie im Thema „Benutzerdefinierte Befehle verwenden“ des NiceLabel Automation Benutzerhandbuchs.

**HINWEIS:** Die meisten benutzerdefinierten Befehle sind in einzelnen Aktionen verfügbar, sodass Sie in den meisten Fällen keine benutzerdefinierten Befehle benötigen.

**HINWEIS:** 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

## 6.4.3 Drucker

### 6.4.3.1 Drucker Einstellen

Diese Aktion gibt den Namen des Druckers an, der zum Drucken des aktiven Etiketts verwendet werden soll.

**HINWEIS:** 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 Ausdrücke 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.

### Ausführen von Aktionen und Fehlerbehebung

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 Programmieraussdruck, 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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.3.2 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.

**HINWEIS:** 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.

**HINWEIS:** Im Modul Automation Builder wird der Variablenwert normalerweise durch einen Filter zugewiesen.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.3.3 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.

**HINWEIS:** 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.

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.

**HINWEIS:** 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.

**HINWEIS:** Diese Aktion leitet den Druck lediglich um. Stellen Sie sicher, dass darauf die Aktion **Etikett drucken** folgt.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.3.4 Druckparameter Festlegen

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.

Wenn Sie die Aktion [Drucker einstellen](#) nutzen, um den Druckernamen zu ändern, müssen Sie die Aktion **Druckparameter festlegen** direkt danach verwenden. 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 der DEwe 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.

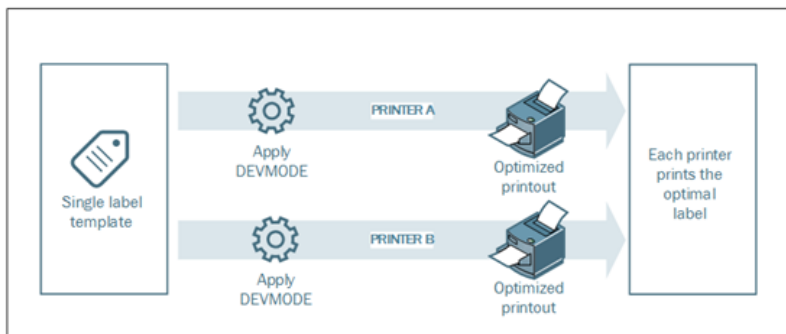
**BEISPIEL:** Das erste Druckermodell akzeptiert Werte zwischen 0 und 30, während das zweite Werte zwischen -15 und 15 akzeptiert. Weitere Informationen finden Sie unter „Druckertreiber-Einstellungen“.

- **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.** 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.

**HINWEIS:** Wenn die ausgewählte Variable keine binäre DEVMODE-Struktur bereitstellt, müssen Sie sicherstellen, dass sie in der Konfiguration als binäre Variable definiert ist.

**HINWEIS:** Stellen Sie sicher, dass vor dieser Aktion die Aktion [Drucker einstellen](#) definiert ist.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.3.5 Druckumleitung An PDF

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 2017 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.

**HINWEIS:** 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).

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen

`ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

### 6.4.3.6 Druckerstatus

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.

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 Übermittlung Aktion an das Hauptsystem, z. B. über [Daten an TCP/IP-Port senden](#), [HTTP-Anfrage](#), [SQL-Anweisung ausführen](#) oder [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 KB189](#).

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**

<b>Spoolerstatus-ID (als Hexadezimalwert)</b>	<b>Spoolerstatus-Beschreibung</b>
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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.3.7 Etikett Im Drucker Speichern

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.

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

## 6.4.4 Variablen

### 6.4.4.1 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.4.2 Variable Daten Speichern

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.4.3 Variable Daten Laden

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

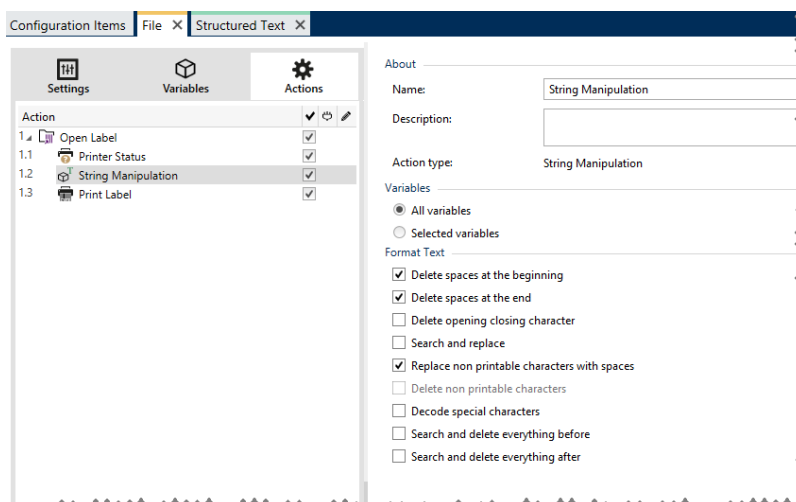
#### 6.4.4.4 Zeichenfolgenmanipulation

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** geparkt werden. Mithilfe der Aktion „Zeichenfolgenmanipulation“ können Sie Anpassungen am Datenwert vornehmen.

**HINWEIS:** 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.

**HINWEIS:** Es gibt verschiedene Implementierungen regulärer Ausdrücke. NiceLabel 2017 nutzt die .NET Framework-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie im [Knowledge Base-Artikel KB250](#).

- **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

2017 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 [[[Undefined variable Variables.Edition-Designer V7]]] 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 neue 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

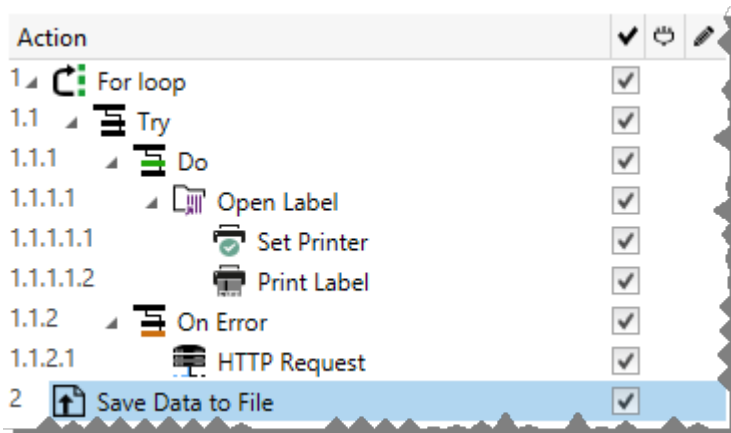
- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

## 6.4.5 Stapeldruck

### 6.4.5.1 FOR Schleife

**DESIGNER-PRODUKTEBENE 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.



**HINWEIS:** 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-Schritt看wert 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.



## 6.4.5.2 Datenfilter Verwenden

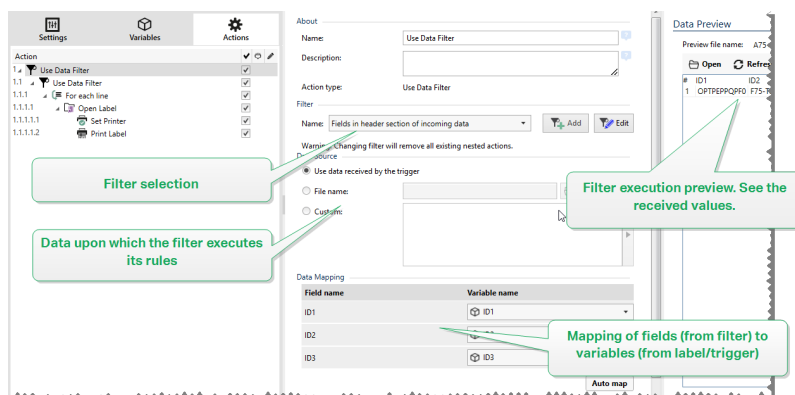
Diese Aktion wendet die Filterregeln auf die Eingabedatenquelle an. Als Ergebnis der Aktion werden Felder aus Eingabedaten extrahiert und ihre Werte den verbundenen Variablen zugeordnet.

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.

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.



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.

**HINWEIS:** 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 geparkt 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 geparkt 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

### 6.4.5.3 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

## 6.4.6 Daten Und Konnektivität

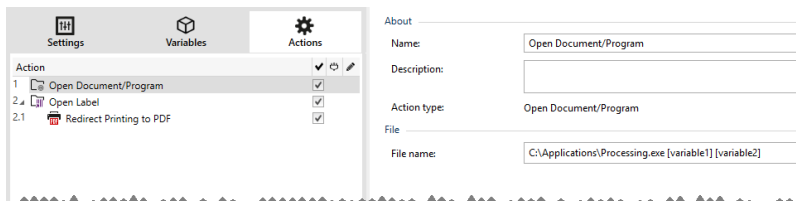
### 6.4.6.1 Dokument/Programm Öffnen

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 2017 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]
```



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.

**HINWEIS:** Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Thema **„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 2017 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.2 Daten In Datei Speichern

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.3 Daten Aus Datei Lesen

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 geparkt 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](#)), während der andere Trigger die Daten liest.

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.

**HINWEIS:** 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.

**HINWEIS:** „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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.4 Datei Löschen

**DESIGNER-PRODUKTEBENE 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.

**HINWEIS:** Verwenden Sie für Netzwerkressourcen die UNC-Syntax. Weitere Informationen finden Sie im Abschnitt „Zugriff auf freigegebene Ressourcen im Netzwerk“ im NiceLabel Automation Benutzerhandbuch.

#### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.5 SQL-Anweisung Ausführen

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.

**HINWEIS:** Um eine Variable in einer SQL-Anweisung zu nutzen, müssen Sie ihrem Namen einen Doppelpunkt (:) voranstellen. So geben Sie an, dass ein Variablenname folgt.

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.

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.

Um die Werte aus den ausgegebenen Daten zu extrahieren und sie in anderen Aktionen zu verwenden, definieren Sie die Aktion „Datenfilter verwenden“ und wenden Sie sie auf den Inhalt dieser Variablen an (diese Aktion ist verfügbar in Automation Builder).

- **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.

**HINWEIS:** Automatische Zuordnung ist aktiviert. Die Aktion „Für jeden Datensatz“ kann nicht gelöscht werden.

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.6 Daten An TCP/IP-Port Senden

**DESIGNER-PRODUKTEBENE INFO**Die beschriebene Funktion ist in **NiceLabel LMS Enterprise** und **NiceLabel LMS Pro** verfügbar.

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.



- **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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.7 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.8 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.9 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.

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 2017 Benutzerhandbuchs.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.10 HTTP-Anfrage

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. Bei dieser Aktion übernimmt NiceLabel 2017 die Rolle eines Clients, der mit einem entfernten 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.

**HINWEIS:** 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.

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 2017 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 2017 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.

HTTP-Header müssen anhand der folgenden Syntax eingegeben werden:

```
Header-Feldname: Header-Feldwert
```

Um beispielsweise die Header-Feldnamen `Accept`, `User-Agent` und `Content-Type` zu verwenden, könnten Sie die folgende Syntax nutzen:

```
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.

**HINWEIS:** Die eingegebenen HTTP-Header heben die Header auf, die bereits anderswo in den Aktionseigenschaften definiert sind, z. B. **Content-Type**.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.6.11 Webdienst

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.

**HINWEIS:** 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.

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:

**HINWEIS:** 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 2017 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 2017 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 Fehlerbehebungs Zwecken 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

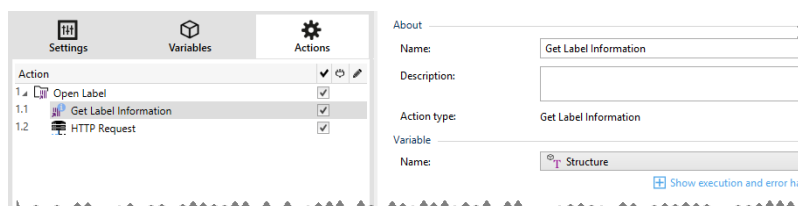
## 6.4.7 Sonstiges

### 6.4.7.1 Etiketteninformationen Abrufen

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.



**HINWEIS:** 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 2017 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.

**HINWEIS:** 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 2017 im Druck-Dialogfeld aufgelistet sind. Gibt es keine Abfragevariablen auf dem Etikett, ist das Element `Variables` leer.
  - **Name**: enthält den Variablennamen.
  - **Description**: 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.

### 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>
```

## Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen *ActionLastErrorId* und *ActionLastErrorDesc* gespeichert.

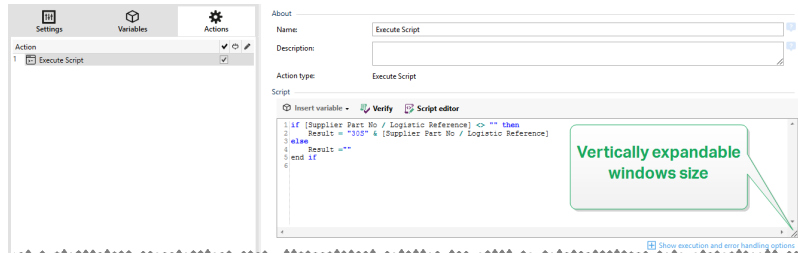
### 6.4.7.2 Script Ausführen

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. Weitere Informationen finden Sie im Abschnitt .



**HINWEIS:** 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

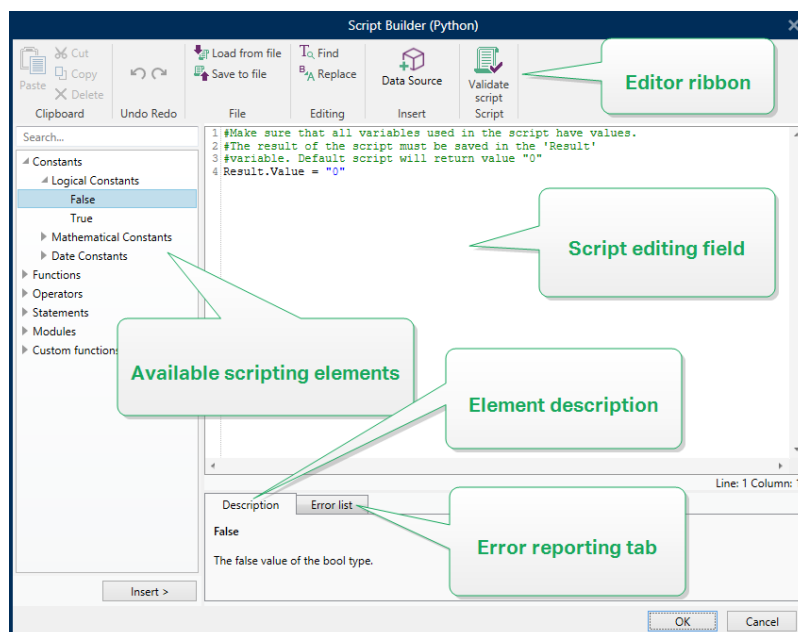
**HINWEIS:** Geschachtelte Aktionen, die von der aktuellen Aktion abhängig sind, werden im Fall eines Fehlers nicht ausgeführt. Die Aktionsausführung wird bei der nächsten Aktion fortgesetzt, die sich auf derselben Ebene wie die aktuelle Aktion befindet. Der Fehler wird protokolliert, führt aber nicht 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen *ActionLastErrorId* und *ActionLastErrorDesc* gespeichert.

### 6.4.7.2.1 Skript-Editor

NiceLabel 2017 bietet einen Skript-Editor, der Ihre Scripting-Aktivitäten mit Python oder VBScript einfacher, fehlerfrei und zeiteffizient macht.



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 externen, 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.

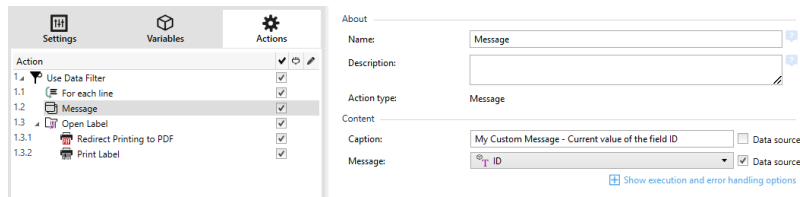
Die **Fehlerliste** enthält die Fehler, die nach Ausführung des Befehls **Skript prüfen** ausgegeben wurden.

#### 6.4.7.3 Meldung (Konfiguration)

Diese Aktion schreibt einen benutzerdefinierten Eintrag in die Protokolldatei.

Normalerweise enthält die Protokolldatei von der Anwendung erzeugte Zeichenfolgen und Fehlerbeschreibungen. Anhand dieser Aktion können Sie eine benutzerdefinierte Zeichenfolge hinzufügen. Dies ist bei der Fehlerbehebung und beim Debugging der Konfiguration nützlich, um die Werte der ausgewählten Variablen nachzuverfolgen.

**BEISPIEL:** Um die Protokollierung von benutzerdefinierten Nachrichten im Protokollbereich von Automation Builder (beim Testen der Konfiguration) oder im Protokollbereich von Automation Manager (nachdem der Trigger implementiert und gestartet wurde) zu konfigurieren, nutzen Sie die folgenden Bildschirmfotos.



Timestamp	ID	Name	Description
14.5.2015 15:14:35.443		Database	Trigger "Database" was stopped.
14.5.2015 15:14:33.121		Database	Trigger was executed - Number of retrieved rows: 1
14.5.2015 15:14:33.122	1	For Each Record action	Action started
14.5.2015 15:14:33.122	1.1	Message.action	My Custom Message - Current value of field ID - 254
14.5.2015 15:14:33.122	1.2	Open Label action	Label: C:\temp\db\label.lbl
14.5.2015 15:14:33.122	1.2.1	Redirect Printing to PDF action	C:\temp\db\labels.pdf
14.5.2015 15:14:33.122	1.2.2	Print Label action	Label: label, Printer: ZEBRA R-402, Quantity: 1
14.5.2015 15:14:33.136	1.2.3	Set Variable action	Set variable "Feedback" to "F".
14.5.2015 15:14:33.136	1.2	Open Label action	Action completed
14.5.2015 15:14:33.136		Execute SQL statement action def...	Action completed - Number of affected rows: 1
14.5.2015 15:14:33.136	1	For Each Record action	Action completed

Die Gruppe **Inhalt** definiert die Titelzeile und den Nachrichteninhalt.

- **Überschrift:** gibt den benutzerdefinierten Meldungstitel an. Die Option **Datenquelle** aktiviert die dynamische Definition des Titels. Wählen Sie eine Variable aus (oder erstellen Sie eine Variable), die nach Ausführen des Triggers den Titel bereitstellt.
- **Meldung:** Gibt den Inhalt der benutzerdefinierten Meldung an. Die Option **Datenquelle** aktiviert die dynamische Definition des Meldungsinhalts.

**TIPP:** Dynamische Inhalte werden normalerweise im Voraus in einer anderen Aktion erstellt und später hier verwendet.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.7.4 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.

**HINWEIS:** Der Lizenzschlüssel, der die Software aktiviert, kann auch die **Lösungs-ID** codieren. Dies ist eine eindeutige Nummer, die den Lösungsanbieter identifiziert, der die NiceLabel 2017-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.

**HINWEIS:** Auch die Digital Partner UID kann als Lizenz-ID verwendet werden. Diese Option ist für Mitglieder des [NiceLabel Digital Partner Programms](#) verfügbar.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen *ActionLastErrorId* und *ActionLastErrorDesc* gespeichert.

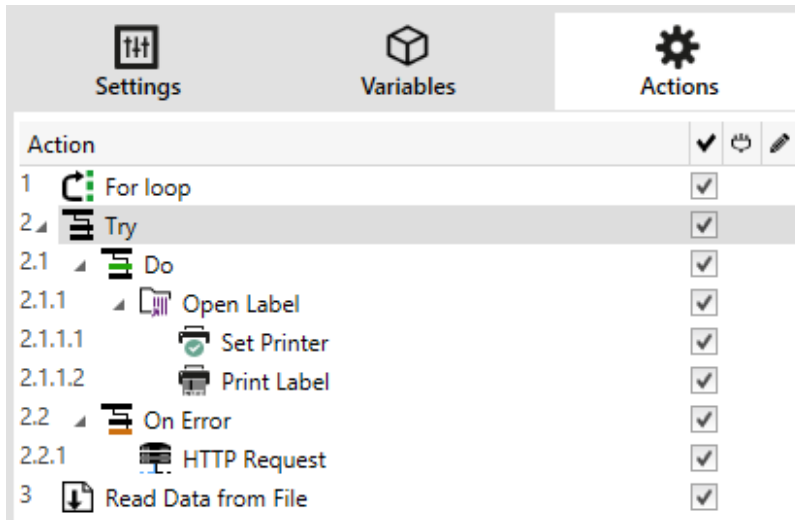
#### 6.4.7.5 Testen

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.



**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.

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.

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

### 6.4.7.6 XML-Umwandlung

**DESIGNER-PRODUKTEBENE INFO**Die beschriebene Funktion steht in **NiceLabel 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](#).

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 ein XML-Dokument in nativ unterstütztes XML konvertieren, z. B. Oracle XML, und es danach einfach anhand der Aktion [Oracle XML-Befehlsdatei ausführen](#) ausführen.

**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.mixx 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.

Die 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.7.7 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.

#### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen *ActionLastErrorId* und *ActionLastErrorDesc* gespeichert.

#### 6.4.7.8 Ereignis Protokollieren

Diese Aktion protokolliert ein Ereignis zu Verlaufs- und Fehlerbehebungs-Zwecken in NiceLabel Control Center.

**HINWEIS:** 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](#).

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen *ActionLastErrorId* und *ActionLastErrorDesc* gespeichert.

### 6.4.7.9 Etikettenvorschau

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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

#### 6.4.7.10 Etikettenvariante Erstellen

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 die enthaltenen Objekte anzuzeigen, genehmigt die zuständige Person eine Variante mit definierten Werten. So kann sie das finale Etikettenlayout mit den tatsächlichen, für den Druck verwendeten Werten, 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 2017 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 (Variante) 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, für die der Wert zum tatsächlichen Druckzeitpunkt bereitgestellt wird. 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 Charge, 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 ihren Wert zum Druckzeitpunkt festlegen.

2. Zählervariablen, Funktionen, Datenbankfelder und globale Variablen werden in nicht-abgefragte Variablen konvertiert.
3. Grafiken werden eingebettet.
4. Die im NiceLabel Control Center Dokumentenspeicher 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 2017 Designer geöffnet werden.
6. Anhand dieser Aktion erzeugte Etikettendateien können nicht importiert werden.
7. Wenn eine solche Etikettenvariante im Drucker gespeichert wird, 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.

### Ausführen von Aktionen und Fehlerbehebung

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.

**HINWEIS:** 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.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. Dieselbe Fehlerbeschreibung wird zudem in den internen Variablen `ActionLastErrorId` und `ActionLastErrorDesc` gespeichert.

## 6.5 Trigger Testen

### 6.5.1 Trigger Testen

Nach Abschluss der Konfiguration des Triggers ist die Arbeit noch nicht beendet. Bevor Sie ihn implementieren, müssen Sie den Trigger durch Anwendung auf eingehende Daten sorgfältig testen, um sicherzustellen, dass er zum gewünschten Ergebnis führt.

Sie können Ihre Konfiguration testen, während Sie noch in Automation Builder daran 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, wobei die Trigger-Ausführung in Automation Manager überwacht wird.

#### Ausführung der einzelnen Aktionen testen

Einige der Aktionen verfügen über Vorschaufunktionen, mit denen Sie die Eingabeparameter ändern und das Ergebnis der Aktion auf dem Bildschirm anzeigen können.

- **Datenfilter verwenden.** Die Aktion zeigt eine Echtzeitvorschau der gepushten Daten an. Die Regeln aus dem ausgewählten Filter werden auf die ausgewählten Eingabedaten angewandt und das Ergebnis in der Tabelle angezeigt. Wenn Sie Unter- oder Zuweisungsbereiche verwenden, können Sie die Vorschau für jede Ebene der Filterdefinition anzeigen.
- **SQL-Anweisung ausführen.** Die Aktion zeigt eine Echtzeitvorschau der Ausführung der definierten SQL-Anweisung an. Darin sehen 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 und Sie können alle Änderungen rückgängig machen. Sie können die Abfrageparameter ändern und erkennen, inwiefern sich die Änderung auf das Ergebnis auswirkt.
- **Webdienst.** Die Aktion zeigt eine Vorschau der Ausführung der 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 Änderungen des überwachten Ereignisses nicht aus; dies ist nur bei Triggern möglich, die im Automation Manager gestartet werden. Stattdessen führt der Trigger Aktionen auf Basis von in einer Datei gespeicherten Daten aus. Sie müssen sicherstellen, 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, 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.

**HINWEIS:** Wenn Sie die Vorschau ausführen, werden alle für den ausgewählten Trigger definierten Aktionen ausgeführt, nicht nur die jeweilige Aktion. 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 können weitere Konfigurationsprobleme erkannt werden, die beim Testen des Triggers im Automation Builder nicht bemerkbar sind. 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 dann 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 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 folgenden Problemen kommen, wenn sich Plattformunterschiede (32-Bit gegenüber 64-Bit) auf die Trigger-Verarbeitung auswirken:
  - **Datenbankzugriff.** 64-Bit-Anwendungen erfordern 64-Bit-Datenbanktreiber, während 32-Bit-Anwendungen 32-Bit-Datenbanktreiber benötigen. 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 Dateien im Netzwerk verwenden.** 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](#).

## 6.6 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.

## 6.7 Firewall Für Netzwerk-Trigger Konfigurieren

Ein Netzwerk-Trigger ist ein Trigger, der anhand des TCP/IP-Protokolls ausgeführt wird. In Automatisierung 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-Trigger 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 Automatisierung Dienst belegten Ports von der Windows Firewall zugelassen werden.

**WARNUNG:** Der Automatisierung 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 Automatisierung Dienstprozess.

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.

## 6.8 Sicheres Übertragungsprotokoll (HTTPS) Nutzen

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Enterprise** zur Verfügung.

Sie können den beim [HTTP Server Trigger](#) und beim [Webdienst-Trigger](#) eingehenden 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“.

**HINWEIS:** 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.
  1. Öffnen Sie ein Eingabeaufforderungsfenster.
  2. Geben Sie **mmc** ein und drücken Sie die Eingabetaste (Sie müssen die Anwendung mit Administratorrechten ausführen).
  3. Wählen Sie im Menü „Datei“ **Snap-In hinzufügen/entfernen**.
  4. Wählen Sie in der Liste **Verfügbare Snap-Ins** die Option **Zertifikate** aus.
  5. Klicken Sie auf **Hinzufügen**.
  6. Wählen Sie im Dialogfeld **Zertifikat-Snap-In** die Option **Computerkonto** und klicken Sie auf **Weiter**.
  7. Klicken Sie im Dialogfeld **Computer auswählen** auf **Fertig** stellen.
  8. Klicken Sie im Fenster **Snap-Ins hinzufügen** bzw. entfernen auf **OK**.
  9. Erweitern Sie im Konsolenstamm-Fenster **Zertifikate > Eigene Zertifikate**.
  10. Klicken Sie mit der rechten Maustaste auf den Zertifikatorbner und wählen Sie **Alle Aufgaben > Importieren**.
  11. Folgen Sie den Schritten im Assistenten, um das Zertifikat zu importieren.
3. Rufen Sie den Fingerabdruck des Zertifikats ab, das Sie soeben importiert haben.
  1. Doppelklicken Sie im MMC auf das Zertifikat.
  2. Öffnen Sie im Dialogfeld **Zertifikat** die Registerkarte **Details**.
  3. Scrollen Sie durch die Liste von Feldern und klicken Sie auf „Fingerabdruck“.
  4. 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=7866c25377554ca0cb53bcd5ee23ce895bdfa2 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` der Fingerabdruck (SHA-Hash) des Zertifikats ist. 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:

1. 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` das IP-Adresse/Port-Paar ist, an dem der Trigger ausgeführt wird und an das Sie das Zertifikat gebunden 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.

# 7 Trigger ausführen und verwalten

## 7.1 Konfiguration Anwenden

Wenn Sie die Trigger im Automation Builder konfiguriert und getestet haben, müssen Sie die Konfiguration im NiceLabel Automation-Dienst anwenden und die Trigger starten. Daraufhin werden die Trigger aktiv und beginnen mit der Überwachung festgelegter 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.

Falls diese Konfiguration bereits geladen war, wird die Anwendung ein erneutes Laden erzwingen, wobei der aktive Status der Trigger aufrechterhalten bleibt.

### 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 zu der gespeicherten Konfiguration.
4. Starten Sie die Trigger, die Sie aktivieren möchten.

### Implementierung aus der Befehlszeile

Um die Konfiguration `C:\Project\Configuration.MISX` anzuwenden und den darin enthaltenen Trigger namens `CSVTrigger` 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 Kapitel [Dienst mit Befehlszeilenparametern steuern](#).

## 7.2 Ereignisprotokoll-Optionen

**WARNUNG:** Einige der in diesem Thema beschriebenen Funktionen erfordern den Kauf von **NiceLabel LMS**-Produkten.

NiceLabel Automation protokolliert Ereignisse an verschiedenen Zielorten, welche vom Implementierungs-Szenario abhängen. Die ersten beiden Protokollfunktionen sind in jedem NiceLabel Automation-Produkt verfügbar.

- **Protokollierung in der Log-Datenbank.** Die Protokollierung in der Log-Datenbank ist immer aktiviert und protokolliert alle Ereignisse und alle Details. Bei der Anzeige der protokollierten Informationen können Sie Filter verwenden, um Ereignisse anzuzeigen, die den Regeln entsprechen. Weitere Informationen finden Sie im Abschnitt [Ereignisprotokoll verwenden](#).

Die Daten werden in der 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. Die 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 eine SQLite-Verwaltungssoftware eines Drittanbieters.

- **Protokollierung im Windows-Anwendungsereignisprotokoll.** Wichtige Ereignisse werden im Windows-Anwendungsereignisprotokoll gespeichert, damit Sie eine zweite Ressource für protokollierte Ereignisse haben, falls NiceLabel Automation nicht gestartet werden kann.
- **Protokollierung im Control Center.** Die Protokollierung im Control Center ist in **NiceLabel LMS Enterprise** und **NiceLabel LMS Pro** Produkten verfügbar. Das Control Center ist eine webbasierte Management-Konsole, die alle Ereignisse von 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.

**HINWEIS:** Weitere Informationen finden Sie im Handbuch zu Control Center.

## 7.3 Trigger Verwalten

Die Anwendung Automation Manager ist der Verwaltungsteil der NiceLabel Automation-Software. Wenn Sie Automation Builder zur Konfiguration der Trigger verwenden, nutzen Sie Automation Manager, um sie anzuwenden und in der Produktionsumgebung 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 ändern. 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 Konfigurationen nach Status sortiert angezeigt. Wenn Sie die Anzeige **Nach Konfigurationen** aktivieren, werden Trigger aus der ausgewählten Konfiguration unabhängig von ihrem Status gemeinsam angezeigt. 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 Informationen, 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.

**HINWEIS:**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, müssen Sie die Konfiguration neu laden.

### **Konfiguration laden**

Um die Konfiguration zu laden, klicken Sie auf die **Hinzufügen**-Schaltfläche (+) und navigieren zur Konfigurationsdatei (.MISX). Die 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 der 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**

Wenn Sie die Konfiguration im Automation Builder aktualisieren und speichern, werden die Änderungen nicht automatisch im Automation Manager angewendet. Um die Konfiguration neu zu laden, klicken Sie mit der rechten Maustaste auf den Konfigurationsnamen und wählen Sie **Konfiguration neu laden**. Alle Trigger werden neu geladen. Falls Sie die [Zwischenspeicherung von Dateien](#) aktiviert haben, wird durch das erneute Laden die Synchronisierung aller von den Triggern verwendeten Dateien erzwungen.

### **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 zu stoppen, klicken Sie auf die Schaltfläche **Stopp**. Sie können weitere Trigger aus derselben Konfiguration auswählen und alle davon gleichzeitig starten/anhalten.

Außerdem können Sie das Starten/Anhalten über die Befehlszeile steuern. Weitere Informationen finden Sie im Kapitel [Dienst mit Befehlszeilenparametern steuern](#).

### **Trigger-Konflikte handhaben**

Trigger können aufgrund der folgenden Umstände Fehler aufweisen. Sie können solche Trigger erst starten, nachdem 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 einem anderen Trigger.** Zwei Trigger können nicht ein und dasselbe Ereignis überwachen.

**BEISPIEL:** Zwei Datei-Trigger können nicht dieselbe Datei überwachen, und zwei HTTP-Trigger können keine Daten an ein und derselben Schnittstelle empfangen. Falls sich eine Trigger-Konfiguration mit einem anderen Trigger überschneidet, wird der zweite Trigger nicht ausgeführt, 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 weist einen Fehlerstatus auf und die Ereignisdetails werden in der Protokolldatenbank erfasst. Selbst wenn die 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 in 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 verwenden](#).

## 7.4 Ereignisprotokoll Verwenden

Alle Aktivitäten in der NiceLabel Automation Software 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 übereinstimmen.

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.

**HINWEIS:**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 aus der ausgewählten Konfiguration.
- **Logging-Zeitraum.** Legt den Zeitrahmen fest, in dem Ereignisse eingetreten sind. Die Standardeinstellung ist **Letzte 5 Minuten**.
- **Ereignisebene.** Gibt den Typ (Bedeutung) der Ereignisse an, die Sie anzeigen möchten. **Fehler** ist ein Ereignistyp, der die Ausführung verhindert. **Warnung** ist ein Ereignistyp, bei dem Fehler passieren, aber per Konfiguration ignoriert werden sollen. **Information** ist ein Ereignistyp, der alle Informationen protokolliert, die nicht mit Fehlern in Zusammenhang stehen. Die Protokollebene kann in den Optionen konfiguriert werden.
- **Filter 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:** Nutzen Sie diese Option mit Bedacht, da die Datenbank nicht wiederhergestellt werden kann. Dadurch werden **ALLE** protokollierten Ereignisse aus der Datenbank gelöscht, und die Option gilt für alle Trigger, nicht nur für den aktuellen.

# 8 Performance- und Feedback-Optionen

## 8.1 Parallele Verarbeitung

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

Die NiceLabel Automation-Produktreihe wurde entwickelt, um parallele Verarbeitung sowohl bei der eingehenden als auch bei der ausgehenden Verarbeitung zu unterstützen. Dies sorgt auf jedem System, auf dem die Software installiert wird, für maximale Effizienz. NiceLabel Automation kann viele Aufgaben gleichzeitig ausführen und dabei dennoch die Reihenfolge aufrechterhalten, in der die Trigger eingegangen sind. Der Durchsatz an Etikettenaufträgen hängt dabei in hohem Maße von der Hardware ab, auf der die Software ausgeführt wird.

### Eingehende parallele Verarbeitung

Sie können mehrere Trigger auf demselben Rechner ausführen, und sie alle werden gleichzeitig auf Änderungen der überwachten Ereignisse reagieren. Jeder Trigger speichert die Daten aus seinen nicht verarbeiteten Ereignissen in der Warteschlangenliste. Diese Liste puffert eingehende Daten für den Fall, dass zu diesem Zeitpunkt keiner der Druckprozesse verfügbar ist. Sobald der erste Druckprozess verfügbar wird, wird der erste Auftrag anhand des FIFO-Prinzips (First In, First Out) aus der Warteschlange genommen. Dies sorgt für eine richtige Reihenfolge bei der Verarbeitung der eingehenden Daten. Es gewährleistet jedoch nicht, dass das FIFO-Prinzip auch beim Drucken angewandt wird. Siehe nächster Absatz.

**HINWEIS:** Sie können nicht nur mehrere Trigger parallel ausführen, sondern 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, wobei eine Konfiguration nach Dateimaske möglich ist.

### Ausgehende parallele Verarbeitung

Normalerweise ist das Ergebnis des Triggers ein Etikettendruckvorgang. Sie möchten vom Trigger empfangene Daten verwenden und auf die Etiketten drucken. Der NiceLabel Automation-Dienst führt 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, was die allgemeine Verarbeitungsgeschwindigkeit steigert; im Fall von NiceLabel Automation steigern sie die Geschwindigkeit der Verarbeitung von Druckaufträgen und damit auch die Etikettendruck-Performance.



Standardmäßig führt jedes NiceLabel Automation-Produkt Druckprozesse in einem separaten Thread auf jedem Kern aus, der auf dem Rechner zur Verfügung steht. Je leistungsstärker Ihre CPU ist, desto höher ist der verfügbare Durchsatz. So wird die verfügbare CPU-Leistung bestmöglich genutzt. Bei der Installation der Software werden Standardeinstellungen verwendet, sodass jeder verfügbare Kern einen Thread für die Druckverarbeitung übernimmt; unter normalen Umständen müssen Sie diese Einstellungen nicht ändern. Falls Sie eine Änderung vornehmen möchten, finden Sie weitere Informationen unter [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 angewandt wird, siehe Abschnitt [Synchroner Druckmodus](#).

## 8.2 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.** Das Arbeitsspeichercache legt die verwendeten Dateien im Arbeitsspeicher ab. Etiketten, die mindestens einmal verwendet wurden, werden ins Arbeitsspeichercache geladen. Wenn der Trigger den Druck solcher Etiketten anfordert, stehen sie sofort für den Druckprozess zur Verfügung. Das Arbeitsspeichercache ist standardmäßig aktiviert. Der Inhalt des Arbeitsspeichercaches wird für eine bestimmte Konfiguration gelöscht, wenn Sie diese 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.

**HINWEIS:** Wenn ein Etikett 8 Stunden lang nicht genutzt wird, wird es aus dem Zwischenspeicher entfernt.

- **Permanenter Zwischenspeicher.** Der permanente Zwischenspeicher speichert Daten auf der Festplatte und dient zur mittelfristigen Speicherung von Dateien. Das Zwischenspeichern wird pro Dateiojekt verwaltet. Wenn eine Datei aus der Netzwerkfreigabe angefordert wird, 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 in den Optionen einstellen.

### Zeit bis zur Entfernung von Etiketten aus dem Zwischenspeicher verlängern

Nach der ersten Nutzung des Etiketts wird es in den Zwischenspeicher geladen und steht danach für sofortiges Drucken zur Verfügung. Alle Etiketten, die 8 Stunden lang 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 Systemordner von NiceLabel Automation.

```
%PROGRAMDATA%\NiceLabel\NiceLabel 2017
```

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 legt (in Sekunden) fest, wie lange das Etikett im Zwischenspeicher verbleiben soll. NiceLabel Automation speichert die verstrichene Zeit nach dem letzten Drucken des Etiketts. Sobald der festgelegte Schwellenwert erreicht ist, wird das Etikett aus dem Zwischenspeicher entfernt.

**HINWEIS:** Standardwert: 28800 (8 Stunden). Der maximale Wert beträgt 2147483647.

Die Datei sollte den folgenden Inhalt haben:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <Common>
    <FileUpdater>
      <PurgeAge>28800</PurgeAge>
    </FileUpdater>
  </Common>
  ...
</configuration>
```

6. Wenn Sie die Datei speichern, wendet der NiceLabel Automation-Dienst die Einstellung automatisch an.

### Permanenter Zwischenspeicher aktivieren

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation 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 in der Zwischenzeit nicht genutzten Dateien aus dem Zwischenspeicher entfernt werden.

NiceLabel Automation nutzt das folgende lokale Verzeichnis als Zwischenspeicher für entfernte Dateien:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 2017\FileCache
```

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

### Neues Laden des Inhalts des Zwischenspeichers erzwingen

NiceLabel Automation aktualisiert den Inhalt des Zwischenspeichers nach dem festgelegten Zeitintervall (Standardwert sind 5 Minuten).





Um ein neues 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**.

## 8.3 Fehlerhandhabung

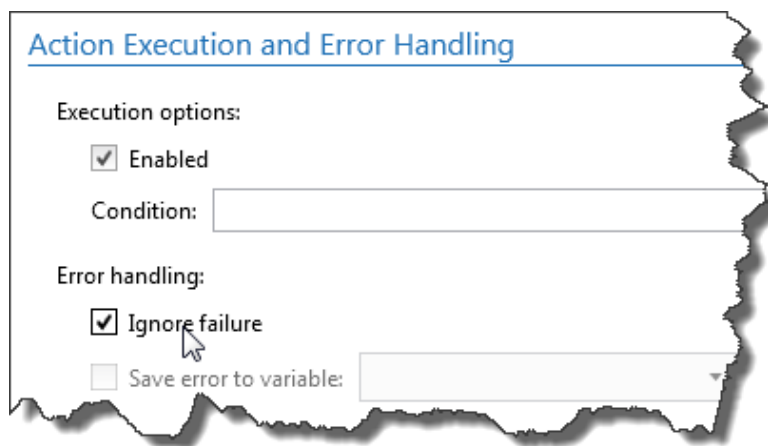
Tritt während der Ausführung einer Aktion ein Fehler auf, beendet NiceLabel Automation die Ausführung aller Aktionen im Trigger. Falls Sie Aktionen nach der aktuellen Aktion definiert haben, werden diese nicht ausgeführt.

Ein Beispiel: Die Aktionen sind gemäß dem Bildschirmfoto definiert. Schlägt die Aktion **Drucker einstellen** fehl, weil ein ungültiger Name oder ein nicht verfügbarer Drucker angegeben wurde, werden die Aktionen **Etikett drucken** und **HTTP-Anfrage** nicht ausgeführt. Die Aktionsverarbeitung bricht bei **Drucker einstellen** ab, Automation Manager zeigt den fehlerhaften Trigger an, und das Status-Feedback des Triggers (falls aktiviert) lautet etwa wie folgt: „falscher Drucker angegeben / Drucker nicht verfügbar“.

- 1  Open Label
- 1.1  Set Printer
- 1.2  Print Label
- 2  HTTP Request

In diesem Fall jedoch wollen Sie kein synchrones Feedback verwenden (wird automatisch gesendet, wenn es für einen Trigger aktiviert wurde, der synchrones Feedback unterstützt). Das Status-Feedback muss asynchron mit der Aktion **HTTP-Anfrage** bereitgestellt werden, nachdem der Druckauftrag erstellt wurde (oder nicht erstellt wurde). Wenn der Druckprozess abgeschlossen ist, wollen Sie eine Anwendung mit seinem Status aktualisieren. Dazu senden Sie eine HTTP-formatierte Meldung an diese Anwendung.

In diesem Fall muss die Aktion **HTTP-Anfrage** unabhängig vom Erfolg aller Aktionen erfolgen, die weiter oben in der Liste definiert sind. Sie müssen die Option **Fehler ignorieren** für alle Aktionen über der Aktion **HTTP-Anfrage** aktivieren. Die Option steht in den Ausführungs- und Fehlerhandhabungs-Optionen der Aktion zur Verfügung.



Schlägt eine bestimmte Aktion fehl, beginnt NiceLabel Automation mit der Ausführung der nächsten Aktion in der nächsthöheren Hierarchiestufe.

**BEISPIEL:** Falls die Aktion **Drucker einstellen** in Stufe 1.1 fehlschlägt, fährt die Ausführung nicht mit der Aktion **Etikett drucken** in Stufe 1.2 fort, da diese wahrscheinlich auch fehlschlagen würde, sondern mit der Aktion **HTTP-Anfrage** in Stufe 2, da dies die nächste Aktion in der nächsthöheren Hierarchiestufe ist.

Dieselbe Logik kann für die Schleifenausführung von Aktionen verwendet werden, zum Beispiel **Datenfilter verwenden, Schleife** und **Für jeden Datensatz**, wobei alle Elemente in der Liste schrittweise durchgegangen werden. 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.

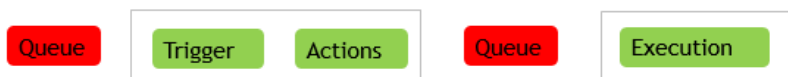
## 8.4 Synchroner Druckmodus

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

### Asynchroner Druckmodus

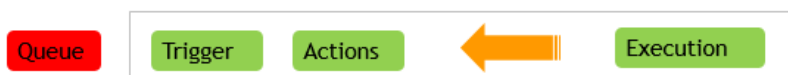
Der Standard-Betriebsmodus von NiceLabel Automation ist der asynchrone Modus. Es handelt sich dabei um eine Art des Druckens, bei der ein Trigger die Druckdaten sendet und danach die Verbindung zum Druck-Subsystem schließt. Der Trigger wartet nicht auf das Ergebnis des Druckprozesses und erhält kein 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. Jedem Druckprozess ist ein Puffer vorangestellt, in den der Trigger die Druckanfragen 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 Control Center, falls es genutzt wird) protokolliert, aber der Trigger selbst erhält kein entsprechendes Feedback. Bei der Verwendung des synchronen Modus können Sie keine von Bedingungen abhängigen Aktionen definieren, die ausgeführt werden, falls bei der Trigger-Ausführung ein Fehler auftritt.



### Synchroner Druckmodus

Der synchrone Druckmodus hingegen unterbricht die Verbindung zum Druckprozess nicht. In diesem Modus sendet der Trigger die 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 andere Aktionen ausgeführt werden sollen. Außerdem können Sie den Druckauftragsstatus an die datengebende Anwendung zurücksenden. Weitere Informationen finden Sie im Abschnitt [Feedback zum Status von Druckaufträgen](#).



**BEISPIEL:** Sie können den Druckstatus an die ERP-Anwendung senden, die die Daten bereitgestellt hat.

Sie sollten den synchronen Modus verwenden, wenn Sie Status-Feedback innerhalb des Triggers erhalten oder sicherstellen wollen, dass das FIFO-Prinzip auf den Druckprozess

angewandt wird (d. h., die in den Trigger-Ereignissen empfangenen Daten werden in der Reihenfolge ihres Eingangs gedruckt).

**HINWEIS:** Wird der Trigger im synchronen Druckmodus ausgeführt, kommuniziert er nur mit einem 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 synchrone Modus 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 die Option **Andere**.
4. Aktivieren Sie im Bereich **Feedback von der Print Engine** die Option **Überwachtes Drucken**.

## 8.5 Feedback Zum Status Von Druckaufträgen

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht nicht in jedem NiceLabel Automation-Produkt zur Verfügung.

Die Anwendung, die Daten für den Etikettendruck in NiceLabel Automation bereitstellt, erwartet möglicherweise Informationen zum 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 deaktiviert. So wird ein hoher Druckdurchsatz gewährleistet, da der Trigger sich nicht um die Ausführung des Druckprozesses 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 entgegennehmen kann, etwa zum Status der Netzwerkdrucker, der Anzahl von Aufträgen im Drucker-Spooler, einer Liste von Etiketten in einem Ordner, der Liste von Variablen in der angegebenen Etikettendatei und vielen mehr.

**HINWEIS:** Um die Feedback-Unterstützung in der Druck-Engine zu aktivieren, müssen Sie den synchronen Druckmodus aktivieren. Weitere Informationen finden Sie im Abschnitt [Synchroner Druckmodus](#).

Es gibt zwei Methoden, um Status-Feedback bereitzustellen.

### 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 nutzen, müssen Sie einen Trigger verwenden, der sie unterstützt.

Tritt in einer der Aktionen ein Fehler auf, enthält die interne Variable `LastActionErrorDesc` die entsprechende ausführliche Fehlermeldung. Sie können ihren Wert 1:1 senden oder ihn anpassen.

Weitere Informationen finden Sie in den Angaben zum jeweiligen Trigger.

- **Webdienst-Trigger.** Dieser Trigger 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 zurücksendet, 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 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. Sie müssen den datengebenden Client dafür konfigurieren, die Verbindung nicht zu trennen, nachdem die Daten gesendet wurden. Nach Abschluss des Druckprozesses kann die nächste Aktion in der Liste die Einstellung **Absender antworten** haben. Das Feedback kann über die bestehende, noch offene Verbindung gesendet werden.

### **Der 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, können Sie eine Aktion definieren, die das Feedback an ein festgelegtes 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 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 alle Aktionen für ausgehende Verbindungen konfigurieren, darunter Daten an HTTP senden, Daten an serielle Schnittstelle senden, Daten an TCP/IP Port senden und andere. Solche Aktionen werden unter der Aktion 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 definieren.** Der letzte Status eines Druckauftrags ist in zwei internen Variablen enthalten (`LastActionErrorID` und `LastActionErrorDesc`). Die

erste enthält die Fehler-ID oder, falls keine Fehler aufgetreten sind, den Wert 0. Die zweite enthält die ausführliche Fehlermeldung. Sie können die Werte dieser Variablen in Bedingungen für Aktionen verwenden, die im Fall von Fehlern ausgeführt werden sollen. Beispielsweise würden Sie die Aktion **HTTP-Anfrage** nach dem Drucken verwenden und Feedback nur senden, falls ein Fehler aufgetreten ist. Dazu müssten Sie Folgendes tun:

1. Öffnen Sie die Trigger-Eigenschaften.
2. Klicken Sie in der Gruppe „Variable“ in der Multifunktionsleiste auf die Schaltfläche **Interne Variablen** und aktivieren Sie die Variable `LastActionErrorID`.
3. Öffnen Sie die Registerkarte „Aktionen“.
4. Fügen Sie die Aktion **Daten an HTTP senden** hinzu.
5. Erweitern Sie in den Eigenschaften der Aktion den Bereich **Optionen für Ausführung und Fehlerhandhabung anzeigen**.
6. Geben Sie für **Bedingung** Folgendes ein: Die Aktion mit dieser Bedingung wird nur ausgeführt, wenn ein Fehler aufgetreten ist und `LastErrorActionID` die Fehler-ID enthält (ein beliebiger Wert größer 0). Standardmäßig werden die Bedingungen mithilfe der VB Script-Syntax angegeben.

```
LastErrorActionID > 0
```

7. Außerdem müssen Sie die Option **Fehler ignorieren** für jede Aktion aktivieren, die fehlschlagen könnte. So weisen Sie Automation an, die Ausführung der Aktionen nicht vollständig zu beenden, sondern mit der nächsten Aktion auf derselben hierarchischen Ebene fortzufahren.

**HINWEIS:** Weitere Informationen finden Sie im Abschnitt [Fehlerhandhabung](#).

- **Die Aktion „Testen“ verwenden.** Die Aktion „Testen“ macht die Festlegung von Bedingungen überflüssig. Die Aktion bietet Ihnen zwei Platzhalter. Der Platzhalter **Ausführen** enthält die 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 Variablen.

## 8.6 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 die variablen Objekte des Etiketts sowie Abrufbefehle. Der Drucker wendet die empfangenen Daten in den Platzhaltern auf dem gespeicherten Etikett an und druckt das Etikett aus (durch Abruf aus dem internen Speicher). Für gewöhnlich werden nur einige Bytes an Daten an den Drucker gesendet (beim normalen Druck wäre es einige Kilobytes).

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 Etiketten aus dem Etikettendesigner oder aus NiceLabel Automation anhand der Aktion speichern.

**HINWEIS:** In den Eigenschaften des Etiketts muss der Speichern-und-Abrufen-Druck definiert werden, bevor Sie es im Drucker speichern 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, müssen Sie eine der üblichen Druckaktionen nutzen. 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.

## 8.7 Hochverfügbarkeits-(Failover-)Cluster

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation 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. Clients, die Daten bereitstellen, verbinden sich mit der IP-Adresse des Clusters, nicht mit der einzelner Knoten.

So aktivieren Sie die Hochverfügbarkeit in NiceLabel Automation:

- 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>Werkzeuge>Optionen**.
  2. Wählen Sie den Abschnitt **Cluster-Unterstützung**.
  3. Aktivieren Sie die **Failover-Cluster-Unterstützung**.
  4. 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. Die wichtigen 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.

## 8.8 Lastausgleichs-Cluster

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation 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. Clients, die Daten bereitstellen, verbinden sich mit der IP-Adresse des Clusters, nicht mit der einzelner Knoten.

**HINWEIS:** Sie können die TCP/IP-basierten Trigger zusammen mit dem Lastausgleichs-Cluster verwenden; dazu zählen der [TCP/IP Server Trigger](#), der [HTTP Server Trigger](#) Trigger und der [Webdienst-Trigger](#).

So aktivieren Sie den Lastausgleich in NiceLabel Automation:

- Richten Sie die Funktion „Microsoft Load-balancing Clustering“ auf Ihren Windows-Servern ein.
- Installieren Sie NiceLabel Automation auf jedem Knoten.
- Laden Sie auf jedem Knoten dieselben Konfigurationsdateien in Automation Manager.

# 9 Informationen zu Datenstrukturen

## 9.1 Informationen Zu Datenstrukturen

Dieses Kapitel beschäftigt sich mit den grundlegenden Datenstrukturen, die in Automationsszenarios häufig zum Einsatz kommen. Wir müssen die Strukturen lesen, die gewünschten Werte extrahieren und sie auf das Etikett drucken. Jedes der aufgeführten Beispiele wird in den Beispielkonfigurationen verwendet, die mit der Software installiert werden. Weitere Informationen finden Sie im Abschnitt [Beispiele](#).

- [Textdatenbank](#)
- [Zusammengesetzte CSV-Dateien](#)
- [Binärdateien](#)
- [Altdaten](#)
- [Befehlsdateien](#)
- [XML-Daten](#)

## 9.2 Binärdateien

Binärdateien sind Dateien, die nicht nur reinen Text, sondern auch Binärzeichen wie Steuercodes (Zeichen unter ASCII-Code 32) enthalten. Der [Filter für unstrukturierte Daten](#) bietet Unterstützung für Binärzeichen. Sie können Binärzeichen verwenden, um Feldpositionen zu definieren und Feldwerte anzugeben.

Ein typisches Beispiel ist der Datenexport aus einem Altsystem, wobei die Daten für die einzelnen Etiketten durch das Steuerzeichen „Form Feed“ (<FF>) getrennt werden.

### Beispiel

In diesem Fall erfasst der 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^I
^^02^00270
```

```

D11
H15
PE
SE
Q0001
131100000300070001-001-001
1e42055007500500001001019
1322000001502859
W
E
<FF><ESC>%-12345X<ESC>%-12345X@PJL USTATUSOFF
<ESC>%-12345X

```

Weitere Informationen finden Sie im Abschnitt [Beispiele](#).

## 9.3 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 in den Abschnitten [Spezifikationen für Befehlsdateien](#), [Oracle XML-Spezifikationen](#) und [SAP All XML-Spezifikationen](#).

### Beispiel

Das Etikett `label12.nlbl` wird auf dem Drucker `CAB A3 203DPI` gedruckt.

```

LABEL "label12.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 im Abschnitt [Beispiele](#).

## 9.4 Zusammengesetzte CSV-Dateien

Eine zusammengesetzte CSV-Datei ist eine Textdatei mit CSV-Struktur sowie einem mehrzeiligen Header in einer anderen Struktur. Der Inhalt kann nicht mit nur einem Filter geparkt werden. Sie müssen zwei Filter konfigurieren, einen [Filter für strukturierten Text](#) für die Felder im CSV-Abschnitt und einen [Filter für unstrukturierte Daten](#) für Felder im Header-Abschnitt. Unter „Aktionen“ definieren Sie zu diesem Zweck zwei Aktionen und wenden beide Filter auf die empfangenen Daten an.

### Beispiel

Die Daten von Zeile 3 bis zum Ende des Dokuments haben eine CSV-Struktur und werden vom Filter für strukturierten Text geparkt. Die Daten in den ersten beiden Zeilen haben keine bestimmte Struktur und werden vom Filter für unstrukturierte Daten geparkt.

```

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 im Abschnitt [Beispiele](#).

## 9.5 Altdaten

Altdaten bezeichnet einen unstrukturierten oder teilweise strukturierten Export aus Altanwendungen. Die Daten weisen weder eine CSV- noch eine XML-Struktur auf, weswegen Sie den [Filter für unstrukturierte Daten](#) verwenden und die Positionen der gewünschten Felder definieren müssen. Der Filter extrahiert Feldwerte, woraufhin Sie sie auf Etiketten drucken können.

### Beispiel

Es gibt keine Regel für die Struktur. 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
```

```
Antikörper-Screen: Negativ  
Probenaufbewahrung :  
PROBE 24 STUNDEN LANG GÜLTIG, KEINE TRANSFUSIONSANAMNESE VORLIEGEND
```

```
07/11/12 B,31.0001245.E O Rh(D) Pos PHO  
RLUH BT
```

Weitere Informationen finden Sie im Abschnitt [Beispiele](#).

## 9.6 Textdatenbank

Textdatenbank ist ein anderer Begriff für eine Textdatei mit strukturierten Feldern, z. B. CSV (Datei mit durch Kommas getrennten Werten) oder eine Datei 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 Kapitel [Dynamische Struktur aktivieren](#).

### Beispiel

- **Datei mit getrennten Feldern.** Die erste Zeile der Datei kann Feldnamen enthalten, 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.**

```
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 im Abschnitt [Beispiele](#).

## 9.7 XML-Daten

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

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.

### Beispiele

#### Oracle XML

Die Software verfügt über integrierte Unterstützung für Oracle XML. Sie müssen keine Filter zur Extraktion von Daten definieren, sondern nur die integrierte Aktion nutzen. 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 Filtern](#).

```

<?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 Software verfügt über integrierte Unterstützung für NiceLabel-XML. Sie müssen keine Filter zur Extraktion der Daten definieren, sondern nur die integrierte Aktion nutzen. 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](#).

# 10 Referenz und Fehlerbehebung

## 10.1 Typen Von Befehlsdateien

### 10.1.1 Spezifikationen Für Befehlsdateien

Befehlsdateien enthalten Anweisungen für den Druckprozess und werden anhand der NiceLabel-Befehle exprimiert. 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.

### 10.1.2 CSV-Befehlsdatei

Die in einer CSV-Befehlsdatei verfügbaren Befehle sind eine Untermenge der NiceLabel Befehle. Sie können die folgenden Befehle verwenden: **LABEL**, **SET**, **PORT**, **PRINTER** und **PRINT**.

CSV steht für Comma Separated Values (durch Kommas getrennte Werte). In solchen Textdateien werden Werte durch Kommas (,) voneinander getrennt. Die Textdateien können Unicode-Werte enthalten (wichtig für mehrsprachige Daten). Jede Zeile in der CSV-Datei enthält die Befehle für eine bestimmte Etikettendruckaktion.

Die erste Zeile muss die 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.

#### 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 vorangestelltes @ sind Namen von Variablen; sie werden gemeinsam 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 jeweilige Datei zugreifen kann. Weitere Informationen finden Sie im Thema „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 entsprechenden Abschnitt des NiceLabel Automation Benutzerhandbuchs. 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. Der Druckprozess umfasst jeweils ein Etiketten-Set. 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.

### 10.1.3 JOB-Befehlsdatei

JOB-Befehlsdateien sind Textdateien mit NiceLabel Befehlen. Die Befehle werden der Reihe nach 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“.

#### Beispiel Für Eine JOB-Befehlsdatei

Diese JOB-Datei öffnet `label2.nlbl`, legt die Variablen fest und druckt ein Etikett. 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
```

## 10.1.4 XML-Befehlsdatei

Die in einer XML-Befehlsdatei verfügbaren Befehle sind eine Untermenge der NiceLabel-Befehle. Sie können die folgenden Befehle verwenden: **LOGIN**, **LABEL**, **SET**, **PORT**, **PRINTER**, **SESSIONEND**, **SESSIONSTART** und **SESSIONPRINT**. In XML-Dateien wird eine leicht abweichende Syntax verwendet.

Das Stammelement in einer XML-Datei ist `<Nice_Commands>`. Darauf muss das Element `<Label>` folgen, welches das zu verwendende Etikett angibt. Es gibt zwei Methoden, um den Etikettendruck einzuleiten: 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 Sie können den Variablenwert festlegen.

### Beispiel Für Eine XML-Befehlsdatei

Das Beispiel zeigt eine strukturelle Ansicht der Elemente und ihrer Attribute, die Sie in einer XML-Befehlsdatei verwenden können.

```
<nice_commands>  
<label name="label1.nbl">  
  
<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 der XML-Befehle

Dieser Abschnitt enthält die 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 einen benutzerdefinierten Wert festlegen.

- **<Nice\_Commands>**. Dies ist ein Stammelement.
- **<Label>**. Gibt die zu öffnende Etikettendatei an. Ist sie bereits geöffnet, wird sie 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 zu der Etikettendatei angeben. Erforderlich.
- **<Print\_Job>.** Dieses Element enthält Daten für einen Etikettenauftrag. Dieses Element kann innerhalb der Befehlsdatei mehrmals vorhanden sein.
  - **Printer.** 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 Papierbögen drucken, die bereits teilweise bedruckt sind. Weitere Informationen finden Sie im Abschnitt „Beispiele“. 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 „Beispiele“. 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 Element **<Print\_Job>**. 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 in dieser Form 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 Werte der Variablen 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:mstns="http://tempuri.org/XMLSchema.xsd" xmlns:xs="http://www.w3.org/2001/X
MLSchema">
<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>
```

## 10.1.5 Oracle XML-Spezifikationen

Das Oracle XML-Format ist so ausgelegt, dass die XML-Inhalte interpretiert, geparkt und als Etikett gedruckt werden können. Eine XML-Dokumenttypdefinition (DTD) definiert die XML-Tags, die in der XML-Datei verwendet werden. Oracle generiert XML-Dateien gemäß dieser DTD, und die Drittsoftware übersetzt die XML-Datei gemäß der DTD.

Um eine solche Befehlsdatei auszuführen, verwenden Sie die Aktion [Oracle XML-Befehlsdatei ausführen](#).

### XML-DTD

Die folgende XML-DTD wird zur Bildung der XML-Daten für synchrone und asynchrone XML-Formate verwendet; sie definiert die Elemente, die in der XML-Datei genutzt werden, eine Liste ihrer Attribute sowie die 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>
<!ATTLIST label _PRINTERNAME CDATA #IMPLIED>
<!ELEMENT variable (#PCDATA)>
<!ATTLIST variable name CDATA #IMPLIED>
```

### 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

Das Etikett wird mit 1 Kopie gedruckt, der Name des Auftrags im Spooler lautet **Serial**. Der Druckername ist in der XML-Datei nicht angegeben; daher wird das Etikett auf dem Drucker gedruckt, der in der Etikettenvorlage definiert ist.

## 10.1.6 SAP All XML-Spezifikationen

NiceLabel Automation kann als RFID-Gerätecontroller agieren, um RFID-Tags zu codieren und Etiketten zu drucken. Weitere Informationen zu SAP All 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 All XML-Befehlsdatei ausführen](#).

### Beispiel Für SAP All XML

Diese SAP All XML-Datei stellt Daten für ein Etikett bereit (es gibt nur ein `<label>`-Element).

```
<?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">Produkt</Field>
<Field name="PRODUCT_DESCRIPTION">Produktbeschreibung</Field>
</FieldList>
</Item>
</WriteTagData>
</Command>
```

Bei Ausführung dieser SAP All 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

Das Etikett wird mit 1 Kopie gedruckt, der Name des Auftrags im Spooler lautet `Writer_Device2004092916574`. Der Druckername ist in der XML-Datei nicht angegeben; daher wird das Etikett auf dem Drucker gedruckt, der in der Etikettenvorlage definiert ist.

## 10.2 Benutzerdefinierte Befehle

### 10.2.1 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 Informationen finden Sie im Abschnitt „Spezifikationen für Befehlsdateien“.

Sie können den jeweiligen benutzerdefinierten Befehl verwenden, wenn er in Ihrem NiceLabel Automation-Produkt als Aktion zur Verfügung steht.

**HINWEIS:** Beispielsweise können Sie den Befehl **SETPRINTPARAM** ausführen, wenn die Aktion **Druckparameter festlegen** angezeigt wird (Produktstufe Pro und Enterprise).

#### 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. Sie können ihn verwenden, 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]
```

Der Befehl wird implementiert, um den Befehl „An Drucker exportieren“ zu automatisieren, der im Etikettendesigner zur Verfügung steht. Das Etikett wird direkt an den Drucker exportiert und zwecks Offline-Druck im Speicher abgelegt. Benutzer können das Etikett per Tastenbefehl am Drucker oder durch Senden einer Befehlsdatei an den Drucker aufrufen. Dieselbe Funktionalität ist auch anhand der Aktion „Etikett im Drucker speichern“ verfügbar.

**HINWEIS:** Um das zu exportierende Etikett anzugeben, verwenden Sie zuerst den Befehl **LABEL**.

- **ExportFileName.** Der Parameter ist obligatorisch und definiert den Dateinamen eines erzeugten Druckerbefehls.
- **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, wenn Sie den Speichern/Abrufen-Druckmodus aktivieren.

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 zum Offlinedruck finden Sie unter [Speichern/Abrufen-Druckmodus verwenden](#).

## IGNOREERROR

```
IGNOREERROR <on> [, <off>]
```

Gibt an, dass der Druckprozess beim Auftreten eines der folgenden Fehler in der JOB-Datei nicht beendet wird:

- Falsche Variablenname verwendet
- Falscher Wert an Variable gesendet
- Etikett ist nicht vorhanden / Zugriff nicht möglich
- Drucker ist nicht vorhanden / Zugriff nicht möglich

## LABEL

```
LABEL <Etikettename> [, <Druckername>]
```

Dieser Befehl öffnet das zu druckende Etikett. Ist das Etikett bereits geladen, 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>]
```

Speichert eine benutzerdefinierte `Meldung` im Trigger-Protokoll. Falls die Meldung Leerzeichen oder Kommas enthält, müssen Sie den Text in Anführungszeichen (") einschließen.

## PORT

```
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 einen 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.

- **Menge.** Gibt die Anzahl von zu druckenden Etiketten an.
  - **<Anzahl>.** Die angegebene Anzahl von Etiketten wird gedruckt.
  - **VARIABLE.** Gibt an, dass eine Etikettenvariable als *variable Menge* definiert ist und die Anzahl zu druckender Etiketten enthalten wird. Der Variablenwert 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.
- **Überspringen.** 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.

**HINWEIS:** Geben Sie die Mengenwerte als numerische Werte an, nicht als Zeichenfolgen-Werte. Stellen Sie den Wert nicht in Anführungszeichen.

## PRINTER

```
PRINTER <Druckername>
```

Dieser Befehl übergeht den in der Etikettenvorlage definierten Drucker. Falls der Druckername Leerzeichen enthält, müssen Sie ihn in Anführungszeichen setzen (").

Verwenden Sie den Druckernamen, wie er in der Statuszeile in der Etikettendesign-Anwendung angezeigt wird. Druckernamen sind normalerweise identisch mit den Druckernamen unter „Drucker und Faxgeräte“ in der Systemsteuerung; dies ist aber nicht immer der Fall. Wenn Sie Netzwerkdrucker verwenden, wird der Name eventuell in 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 enthält, müssen Sie ihn in Anführungszeichen (" ) setzen.

## SESSIONEND

```
SESSIONEND
```

Dieser Befehl schließt den Druckdatenstrom. Siehe auch **SESSIONSTART**.

**HINWEIS:** `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 Befehls SESSIONEND schließen. Die Bedeutung der Parameter „Menge“ und „überspringen“ ist dieselbe wie im NiceCommand PRINT. Siehe auch **SESSIONSTART**.

- **Menge.** Gibt die Anzahl von zu druckenden Etiketten an.
- **Überspringen.** 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 einzelne Etiketten in separaten Druckaufträgen an den Drucker gesendet. Wenn Sie Etikettendaten für mehrere Etiketten in den Druckdatenstrom integrieren möchten, sollten Sie die Befehle für Sitzungsdruck verwenden. Beginnen Sie mit dem SESSIONSTART-Befehl, gefolgt von einer beliebigen Anzahl von SESSIONPRINT-Befehlen, und verwenden Sie am Ende den SESSIONEND-Befehl.

Nutzen Sie diese Befehle, um den Etikettendruckprozess zu optimieren. Das Drucken von Etiketten über einen einzelnen Druckauftrag geht viel schneller als das Drucken von Etiketten aus mehreren Druckaufträgen.

Sie müssen einige Regeln beachten, damit 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. 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 Erhöhungsschritt für den Zähler sowie die Anzahl von Etiketten vor Änderung des Zählerwerts an.

Falls der 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“ im Missing variable reference Benutzerhandbuch.

**HINWEIS:** 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 der 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"%
```

## 10.3 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 dem 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 Kapitel [Im Dienstmodus ausführen](#).
- **UNC-Notation für Netzwerkfreigaben.** Verwenden Sie beim Zugriff auf eine Datei auf einem Netzlaufwerk die UNC-Syntax (Universal Naming Convention) anstelle der zugeordneten Laufwerksbuchstaben. UNC ist eine Benennungskonvention zur Angabe und Zuordnung von Netzlaufwerken. NiceLabel Automation versucht, die Laufwerksbuchstaben-Syntax automatisch durch die UNC-Syntax zu ersetzen.

**BEISPIEL:** Befindet sich die 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 die Datei im Dokumentenspeicher in Control Center öffnen, können Sie die HTTP-Notation als `http://servername:8080/label.nlbl` oder die WebDAV-Notation als `\\servername@8080\DavWWWRoot\label.nlbl` nutzen.

Weitere Hinweise:



- Das zur Ausführung des NiceLabel Automation Dienstes verwendete Benutzerkonto wird verwendet, um Dateien aus dem Dokumentenspeicher zu beziehen. Dieser Benutzer muss in der Control Center-Konfiguration konfiguriert sein, um Zugriff auf Dateien im Dokumentenspeicher zu erhalten.
- Der WebDAV-Zugriff kann nur mit Windows-Benutzerauthentifizierung in Control Center genutzt werden.

**HINWEIS:**Der Dokumentenspeicher ist in den Produkten **NiceLabel LMS Enterprise** und **NiceLabel LMS Pro** enthalten.

- **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 KB265](#).

## 10.4 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 Automatisierung Datei in NiceLabel Control Center unnötig.

**HINWEIS:**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.

**HINWEIS:** 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.

## 10.5 Zugriff Auf Datenbanken

Wenn NiceLabel Automation Daten aus einer Datenbank abrufen soll, müssen Sie sicherstellen, dass im Windows-System der erforderliche Datenbanktreiber installiert ist. Die 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.

### 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 benötigt. Alle NiceLabel Automation-Komponenten werden als 32-Bit-Anwendungen ausgeführt.

### 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 Automatisierung Dienst als 64-Bit-Prozess ausgeführt. In diesem Fall nutzt er 64-Bit-Datenbanktreiber, um eine Datenbankverbindung herzustellen. Wenn im System, auf dem der Automatisierung Dienst ausgeführt wird, keine 64-Bit-Datenbanktreiber enthalten sind, wird die Datenbankverbindungs-Task an den **NiceLabel Proxy**-Prozess übergeben, der immer als 32-Bit-Prozess ausgeführt wird.

## 10.6 Automatisches Ersetzen Von Schriften

Sie können Ihre Etikettenvorlagen so erstellen, dass Textobjekte für den Druck mit integrierten Druckerschriften formatiert werden. Wird ein solches Etikett jedoch auf einer anderen Art von

Drucker gedruckt, stehen die ausgewählten Schriften möglicherweise nicht zur Verfügung. Der andere Drucker unterstützt vermutlich eine völlig andere Reihe von internen Schriften. Diese Schriften können identisch aussehen, aber unter anderen Namen verfügbar sein.

Ein ähnliches Problem tritt auf, wenn die im Etikett verwendete Truetype-Schrift auf dem Zielrechner, über den NiceLabel Automation Etiketten druckt, nicht verfügbar ist.

NiceLabel Automation kann dafür konfiguriert werden, die Schriften auf Etiketten automatisch durch kompatible Schriften zu ersetzen. Sie können die Schriftzuordnung auf Basis der Schriftnamen konfigurieren. Wenn die Originalschrift nicht auffindbar ist, versucht NiceLabel Automation, die erste verfügbare Ersatzschrift zu verwenden, die in der Zuordnungstabelle definiert ist. Wenn kein geeigneter Ersatz vorhanden ist, wird die Schrift Arial Truetype verwendet.

**HINWEIS:** Wenn Sie die Funktion zum Ersetzen von Schriften konfigurieren, werden die Zuordnungsregeln ausgeführt, sobald der Drucker auf dem Etikett geändert wird.

**WARNUNG:** Die Konfiguration von Schriftersetzung wird beim Upgrade nicht beibehalten. Führen Sie vor der Aktualisierung eine Sicherung durch.

### Schriftzuordnung konfigurieren

So konfigurieren Sie die benutzerdefinierte Schriftzuordnung:

1. Öffnen Sie den Date Explorer und navigieren Sie zum folgenden Ordner:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 2017
```

2. Öffnen Sie die Datei **fontmapping.def** im XML-Editor Ihrer Wahl.
3. Erstellen Sie im Element **FontMappings** ein neues Element mit benutzerdefiniertem Namen.
4. Erstellen Sie innerhalb des neuen Elements mindestens zwei Elemente mit dem Namen **Mapping**.
  - Der Wert des ersten Mapping-Elements muss den Namen der Originalschrift enthalten.
  - Der Wert des zweiten Mapping-Elements muss den Namen der Ersatzschrift enthalten.

**HINWEIS:** Es kann zusätzliche Mapping-Elemente mit neuen Schriftnamen geben. Ist die Ersatzschrift nicht verfügbar, versucht NiceLabel Automation, die nächste zu laden. Wenn keine der Ersatzschriften verfügbar ist, wird stattdessen Arial Truetype verwendet.

### Beispiel für eine Mapping-Konfiguration

In diesem Beispiel werden zwei Zuordnungen konfiguriert.

- Die erste Zuordnung ersetzt **Avery**-Schriften durch passende **Novexx**-Schriften. 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** True-type verwendet.
- Die zweite Zuordnung ersetzt **Avery YT100** durch **Novexx YT104**. Ist diese Schrift nicht verfügbar, wird die Schrift **Zebra 0** verwendet. Wenn diese Schrift ebenfalls nicht verfügbar ist, wird **Arial** True-type 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>
```

## 10.7 Standardeinstellungen Für Multi-Thread-Druck Ändern

**TIPP:** Der in diesem Abschnitt beschriebene Funktionsumfang steht in **NiceLabel Automation Pro** und **NiceLabel Automation Enterprise** zur Verfügung.

Jedes NiceLabel Automation-Produkt kann die Vorteile von Mehrkernprozessoren nutzen. Jeder Kern wird zur Ausführung eines Druckprozesses verwendet. Die Hälfte der Kerne wird zur Verarbeitung gleichzeitiger *normaler* Threads, die andere Hälfte zur Verarbeitung gleichzeitiger Sitzungsdruck-Threads verwendet.

**HINWEIS:** 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 2017\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.

## 10.8 Kompatibilität Mit NiceWatch-Produkten

NiceLabel Automation kann Trigger-Konfigurationen laden, die in einem der NiceWatch-Produkte definiert wurden. In den meisten Fällen können Sie die NiceWatch-Konfiguration in NiceLabel Automation ohne Anpassungen ausführen.

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 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.

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

- **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 in NiceWatch-Produkten stehen in NiceLabel Automation zur Verfügung. Einige sind überhaupt nicht verfügbar, während andere nur unterschiedlich konfiguriert werden. Wenn die MIS-Datei nicht unterstützte Funktionen enthält, wird Ihnen eine Liste dieser Funktionen angezeigt und sie werden aus der Konfiguration entfernt.

In diesem Fall müssen Sie die .MIS-Datei in Automation Builder öffnen und die Kompatibilitätsprobleme beheben. Sie müssen die Funktionen von NiceLabel Automation nutzen, um die entfernte Konfiguration neu zu erstellen.

- **Kompatibilität mit den Etikettenvorlagen.** Falls Ihre vorhandenen Etikettenvorlagen Funktionen enthalten, die von der Druck-Engine in NiceLabel Automation nicht unterstützt werden, sehen Sie eine Fehlermeldung im Log-Bereich. 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.

**HINWEIS:** Weitere Informationen über Kompatibilitätsprobleme mit NiceWatch und Etiketten-Designern finden Sie im [Knowledge Base-Artikel KB251](#).

### **NiceWatch-Konfiguration zur Bearbeitung öffnen**

Sie können die vorhandene NiceWatch-Konfiguration (.MIS-Datei) in Automation Builder öffnen und in Automation Builder bearbeiten. Sie können die Konfiguration nur im .MISX-Format öffnen.

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. Falls die Konfiguration nicht unterstützte Funktionen enthält, werden diese als Liste angezeigt. Sie werden aus der Konfiguration entfernt.

### **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. Im Automation Manager wird der Trigger für die ausgewählte Konfiguration angezeigt. Um den Trigger zu starten, wählen Sie ihn aus und klicken Sie auf die Schaltfläche **Start**.

**HINWEIS:** Falls ein Kompatibilitätsproblem mit der NiceWatch-Konfiguration besteht, müssen Sie sie in Automation Builder öffnen und umkonfigurieren.

## 10.9 Dienst Mit Befehlszeilenparametern Steuern

In diesem Abschnitt finden Sie Informationen zur Nutzung der Befehlszeile, um die Automations-Dienste zu starten und anzuhalten und zu steuern, welche Konfigurationen geladen und welche Trigger aktiviert werden.

**HINWEIS:** Stellen Sie sicher, dass Sie die **Eingabeaufforderung** mit erhöhten Rechten (Administratorrechten) ausführen. Klicken Sie mit der rechten Maustaste auf cmd.exe und wählen Sie **Als Administrator ausführen**.

### Dienste starten und anhalten

Um beide Dienste über die Befehlszeile zu starten, verwenden Sie die folgenden Befehle:

```
net start NiceLabelProxyService2017
net start NiceLabelAutomationService2017
```

Wenn Sie beim Starten des Dienstes die Konfigurationsdatei öffnen möchten, verwenden Sie:

```
net start NiceLabelAutomationService2017 [Konfiguration]
```

Zum Beispiel:

```
net start NiceLabelAutomationService2017 "c:\Project\configuration.MISX"
```

Um Dienste anzuhalten, verwenden Sie die folgenden Befehle:

```
net stop NiceLabelProxyService2017
net stop NiceLabelAutomationService2017
```

### Konfigurationen und Trigger verwalten

Der NiceLabel Automation-Dienst kann mithilfe der Automation Manager-Befehlszeilenparameter gesteuert werden. Die allgemeine Syntax für die Nutzung von Befehlszeilenparametern lautet:

```
NiceLabelAutomationManager.exe COMMAND Konfiguration [TriggerName] [/SHOWUI]
```

**HINWEIS:** Hinweis: Verwenden Sie den vollständigen Pfad zu der Konfigurationsdatei, nicht nur den Dateinamen.

### Konfiguration Hinzufügen

Die angegebene Konfiguration wird im Dienst geladen. Es wird kein Trigger 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 im Dienst neu 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 im Abschnitt [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 mit ihr verbundenen 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 Systemvariablen `errorlevel` 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 NiceLabel LMS Enterprise- oder NiceLabel 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.

## 10.10 Ersetzen Der Datenbank-Verbindungszeile

Eine Konfigurationsdatei für den Automation Service kann Befehle zum Ersetzen der Datenbank-Verbindungszeile enthalten.

Benutzer können den Dienst so konfigurieren, dass bestimmte Teile der Verbindungszeile während der Ausführung des Triggers ersetzt werden. Eine Automation-Instanz kann dieselbe Konfiguration, aber unterschiedliche Datenbankserver für datenbankbasierte Funktionen 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` im Automation-Systemordner festgelegt.

```
%PROGRAMDATA%\NiceLabel\NiceLabel 2017
```

Die Konfigurationsdatei definiert From-To-Paare (alt-neu) in der 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 mit Automation installiert. Sie können sie selbst anhand der Struktur aus dem Beispiel hinzufügen. Dieselben Regeln für Suchen und Ersetzen werden auf alle Trigger angewandt, die im Automation Service auf diesem Rechner ausgeführt werden.

**HINWEIS:** Starten Sie beide Automation Services neu, nachdem Sie die config-Datei zum Automation-Systemordner hinzugefügt haben.

### Beispiel

Der vorhandene Trigger beinhaltet eine Verbindung zum Microsoft SQL-Server `mySQLServer` und der Datenbank `myDatabase`. Sie wollen die Verbindungszeile so aktualisieren, dass die Datenbank `NEW_myDatabase` auf dem Server `NEW_mySQLServer` verwendet wird.

Es müssen zwei Replacement-Elemente 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>
```

## 10.11 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:

- **Manuelle Eingabe der Zeichen**, indem Sie eines der beschriebenen Syntax-Beispiele nutzen:
  - Verwenden Sie die Syntax `<special_character_acronym>`, z. B. `<FF>` für Seitenvorschub, `<CR>` für Schlittenrücklauf oder `<CR><LF>` für Zeilenumbruch.
  - Verwenden Sie die Syntax `<#number>`, z. B. `<#13>` für Schlittenrücklauf oder `<#00>` für Nullzeichen.

Weitere Informationen finden Sie im Abschnitt [Liste mit Steuercodes](#).

- **Einfügen der Zeichen aus der Liste**. 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](#).

## 10.12 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

## 10.13 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 Druckererkennung ist eine Kombination aus dem Namen des Druckertreibers (nicht Druckername), dem Standort des Druckers und der Schnittstelle.

„Einen Drucker nutzen“ bedeutet, dass innerhalb einer Automation Konfiguration eine der im Folgenden aufgeführten Aktionen ausgeführt wurde:

- [Etikett drucken](#)
- [Drucker einstellen](#)
- [Daten an Drucker senden](#)
- [Etikettenvorschau](#)
- [Druckereinstellungen definieren](#)
- [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. So stellen Sie sicher, dass der Drucker jederzeit verfügbar ist.

**WARNUNG:** Wenn Sie die Anzahl an Druckerplätzen im Rahmen Ihrer Lizenz überschreiten, können Sie die Software 30 Tage lang in einem Übergangsmodus nutzen. In diesem Modus wird die Anzahl der erlaubten Drucker vorübergehend auf die doppelte Anzahl der Druckerplätze angehoben, die Sie im Rahmen Ihrer Lizenz erworben haben.

Der Übergangsmodus gibt Ihnen ausreichend Zeit, um Lizenzprobleme zu lösen, ohne dass es zu Ausfällen beim Drucken oder beim Erstellen von Etiketten kommt. Häufig treten solche Situationen ein, wenn Sie die Drucker in Ihrer Umgebung ersetzen, alte und neue Drucker gleichzeitig verwenden oder 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 bei den letzten verwendeten Druckern in der Liste.

**TIPP:** Um mehr über die NiceLabel 2017 Lizenzierung zu erfahren, [lesen Sie das Dokument zu diesem Thema](#).

## 10.14 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:

- 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. Zum Beispiel: 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. So stellen Sie sicher, dass NiceLabel Automation nur genehmigte Etiketten druckt, 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 als 64-Bit-Prozess hat einige Vorteile, z. B. direkte Kommunikation mit dem 64-Bit-Druckerspooter-Dienst unter 64-Bit-Windows. Dadurch wird das bekannte Problem mit SPLWOW64.EXE vermieden, einer Middleware, über die 32-Bit-Anwendungen auf den 64-Bit-Druckerspooter-Dienst zugreifen.
- **Zugriff auf Datenbanken.** 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](#).

**HINWEIS:** 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 den 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 **NiceLabelAutomationService2017.x86.exe** und behalten Sie den vorhandenen Pfad bei.
- Starten Sie den NiceLabel Automation-Dienst neu.

**WARNUNG:** Es empfiehlt sich, den NiceLabel Automation-Dienstmodus zu ändern. Wenn Sie dies sowieso tun, stellen Sie sicher, dass Sie vor der Implementierung in der Produktionsumgebung umfassende Trigger-Tests durchführen.

## 10.15 Suchreihenfolge Für Die Angeforderten Dateien

Wenn NiceLabel Automation versucht, eine angegebene Etiketten- oder Bilddatei zu laden, sucht es an verschiedenen Orten nach dieser Datei.

NiceLabel Automation wird in der folgenden Reihenfolge nach der Datei suchen:

1. Prüfen, ob die Datei an dem durch die Aktion vorgegebenen Speicherort vorhanden ist.
2. Prüfen, ob die Datei im selben Ordner wie die Konfigurationsdatei (.MISX) vorhanden ist.
3. Prüfen, ob die Etikettendatei im Ordner .\Labels (bzw., für Grafikdateien, im Ordner .\Graphics) vorhanden ist.
4. Prüfen, ob die Etikettendatei im Ordner ..\Labels (bzw., für Grafikdateien, im Ordner ..\Graphics) vorhanden ist.
5. Prüfen Sie, 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.

## 10.16 Zugriff Auf Ihre Trigger Sichern

Bei manchen Implementierungen müssen Sie einen gesicherten Zugriff auf Ihre Trigger einrichten. NiceLabel Automation ermöglicht es Ihnen, Sicherheitsmaßnahmen zu aktivieren, sodass ein Zugriff auf Trigger auf vertrauenswürdigen Netzwerkgeräten erfolgen kann. Die Sicherheitskonfiguration hängt von der Art des Triggers ab. Einige der Trigger-Typen ermöglichen an sich eine Konfiguration der Zugriffssicherheit. Für alle Trigger, die auf dem TCP/IP-Protokoll basieren, können Sie außerdem alle Details in der Windows-Firewall definieren.

### Firewall konfigurieren

Wenn Sie TCP/IP-basierte Trigger verwenden, etwa den [TCP/IP Server Trigger](#), den [HTTP Server Trigger](#) oder den [Webdienst-Trigger](#), müssen Sie sicherstellen, dass externe Anwendungen sich mit den Triggern verbinden können. Jeder Trigger wird im NiceLabel Automation-Dienst ausgeführt; der Zugriff auf diesen Dienst wird von der Windows-Firewall reguliert. Eine Firewall ist mit einem Schloss an Ihrer Haustür vergleichbar: Sie hilft dabei, Eindringlinge fernzuhalten.

**HINWEIS:** 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 Ihres Unternehmens 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**.

**HINWEIS:** Wenn Sie NiceLabel Automation mit NiceLabel Control Center-Produkten verbunden haben, 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.

## 10.17 Sitzungsdruck

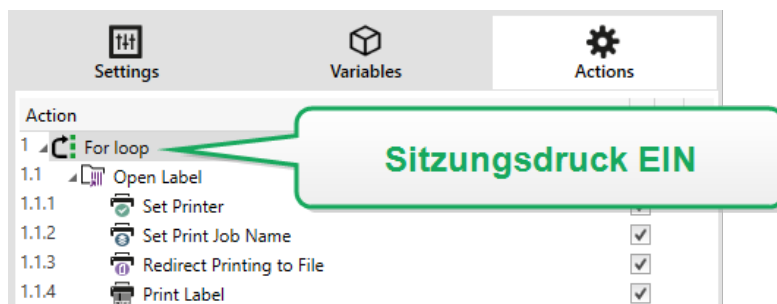
Der Sitzungsdruck ermöglicht es Ihnen, mehrere Etiketten anhand eines einzigen Druckauftrags zu drucken. Wenn der Sitzungsdruck aktiviert ist, empfängt, verarbeitet und druckt der Drucker alle Etiketten im Druckauftrag auf einmal. Aufgrund dieses gebündelten Prozesses steigt die Druckgeschwindigkeit.

**TIPP:** Sitzungsdruck dient als Alternative zum normalen Drucken, bei dem jedes Etikett als separater Druckauftrag an den Drucker gesendet wird.

**HINWEIS:** Automatisierung 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](#) 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.

**HINWEIS:** In diesem Fall muss `SESSIONEND` als einziges Element der Aktion „Benutzerdefinierte Befehle senden“ gesendet werden. Wenn Sie zusätzliche Befehle senden möchten, verwenden Sie separate „Benutzerdefinierte Befehle senden“-Aktionen.

**HINWEIS:** Bei komplexeren Konfigurationen sind möglicherweise mehrere Schleifen ineinander geschachtelt. In solchen Fällen wird der Sitzungsdruck beendet, wenn die äußerste übergeordnete Schleife endet.

## 10.18 Tipps Und Tricks Zur Nutzung Von Variablen In Aktionen

Wenn Sie Variablen in NiceLabel Automation innerhalb von Aktionen nutzen, folgen Sie den hier aufgeführten Empfehlungen.

- **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 können 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 einem [Datenbank-Trigger](#) auf eine Variable zu verweisen, müssen Sie dem Variablennamen einen Doppelpunkt (:) voranstellen, z. B. `:[Produkt ID]`. Der SQL-Parser wird die jeweiligen Daten so als Variablenwert auffassen.

```
SELECT * FROM MyTable WHERE ID = :[ProductID]
```

- **Werte zwecks Berechnung in Ganzzahlen konvertieren.** Wenn Sie numerische Kalkulationen mit den Variablen durchführen wollen, müssen Sie sicherstellen, dass Sie den Variablenwert in eine Ganzzahl konvertieren. Die Definition einer Variable 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 einer Aktion verwenden, achten Sie darauf, dass sie einen Standardwert haben, da ansonsten die Skriptprüfung fehlschlagen könnte. Sie können Standardwerte in den Variableneigenschaften oder innerhalb des Skripts festlegen (und wieder entfernen, nachdem Sie das Skript getestet haben).

## 10.19 Verfolgungsmodus

Standardmäßig speichert NiceLabel Automation Ereignisse in der Protokolldatenbank. Dies umfasst übergeordnete Informationen wie das Protokollieren der Ausführung von Aktionen, der Ausführung von Filtern sowie von Aktualisierungen des Trigger-Status. Weitere Informationen finden Sie im Abschnitt [Ereignisprotokoll-Optionen](#).

Die Standard-Protokollierung zeichnet jedoch die tiefer gehenden Detailinformationen nicht auf. Wenn Fehlerbehebung auf tieferer Ebene im Code notwendig ist, muss der Verfolgungsmodus aktiviert werden. 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 und sollte nur genutzt werden, wenn das technische Support-Team von NiceLabel eine entsprechende Anweisung gibt.

### Aktivierung des Verfolgungsmodus

Um den Verfolgungsmodus zu aktivieren, tun Sie Folgendes:

1. Navigieren Sie zum NiceLabel Automation-Systemordner.

```
%PROGRAMDATA%\NiceLabel\NiceLabel 2017
```

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 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. Wenn Sie die Datei speichern, wendet der NiceLabel Automation-Dienst die Einstellung automatisch an.
6. Die Verfolgungsdateien (\*.LOG) werden standardmäßig im selben Systemordner gespeichert.  
  
Sie können stattdessen einen anderen Ordner im Element `Folder` angeben. Dieses Element ist optional.
7. Um die Aktivierung des Verfolgungsmodus zu bestätigen, starten Sie den Automation Manager. Der Text **Verfolgung wurde aktiviert** wird im Mitteilungsbereich über der Trigger-Liste angezeigt.

## 10.20 Informationen Zu Druckereinstellungen Und DEVMODE

**HINWEIS:** Die DEVMODE-Datenstruktur ist Teil der [GDI Print API-Struktur](#) in Windows. Es handelt sich dabei um hochgradig technische Informationen, die nur unter speziellen Bedingungen benötigt werden.

Wenn Sie ein Etikett in der 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 neuen Druckers auf das Etikett angewandt.

Das Drucken eines Textdokuments auf verschiedenen Laserdruckern führt normalerweise zum selben oder zu einem ähnlichen Ergebnis. Das Drucken von Etiketten auf verschiedenen Etikettendruckern könnte zu unterschiedlichen Ergebnissen führen. Für ein und dieselbe Etikettendatei sind möglicherweise von Druckertreiber zu Druckertreiber zusätzliche Einstellungen erforderlich, um vergleichbare Ergebnisse zu erzielen, etwa eine Anpassung des Versatzes, der Druckgeschwindigkeit oder der Temperatur. Auch NiceLabel wendet die Druckereinstellungen bei jedem Druckvorgang an. Standardmäßig werden die Druckereinstellungen für den ausgewählten Drucker in der Etikettendatei gespeichert.

### Was ist DEVMODE?

DEVMODE ist eine Windows-Struktur, in der die Druckereinstellungen gespeichert sind (Initialisierungs- und Umgebungsinformationen zu einem Drucker). Sie besteht aus zwei Teilen: 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 Hersteller-spezifischen 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 ist 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 die 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 wird von den Treiberentwicklern erstellt und sieht aus wie ein Strom von Binärdaten.

### DEVMODE ändern

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) ändern, indem Sie die Parameter im Druckertreiber ändern. Die ersten beiden Optionen sind Windows-bezogen, die dritte steht in der NiceLabel-Software zur Verfügung.

- **Die Standard-Druckereinstellungen.** Diese sind unter **Druckereinstellungen > Registerkarte „Erweitert“ > Druckstandards** definiert.
- **Die benutzerdefinierten Einstellungen.** Sie werden für jeden Benutzer in dessen HKEY\_CURRENT\_USER-Registrierungsschlüssel gespeichert. Standardmäßig werden die benutzerspezifischen Einstellungen von den Drucker-Standardinstellungen übernommen. Die benutzerspezifischen Einstellungen sind unter **Druckereigenschaften > Einstellungen** definiert. Alle hier vorgenommenen Änderungen betreffen nur den aktuellen Benutzer.
- **Die Etiketten-spezifischen Einstellungen.** Designer, die NiceLabel NiceLabel-Software nutzen, können die DEVMODE-Struktur auch in das Etikett selbst einbetten. Auf diese Weise können die Druckereinstellungen zwischen Rechnern ausgetauscht werden: Beim Kopieren des Etiketts werden die Druckereinstellungen einfach mit übernommen. Um Druckereinstellungen in ein Etikett einzubetten, aktivieren Sie die Option **Im Etikett gespeicherte benutzerdefinierte Druckereinstellungen verwenden** unter *Datei > Etiketteneinrichtung > Registerkarte „Drucker“* Designer Pro in . Sie können die auf dem Etikett eingebetteten Druckereinstellungen ändern, indem Sie *Datei > Druckereinstellungen* auswählen.

### Benutzerdefinierte DEVMODE-Struktur auf den Ausdruck anwenden

In NiceLabel Automation können Sie eine Etikettendatei öffnen und den benutzerdefinierten DEVMODE darauf anwenden. Beim Drucken des Etiketts wird das Etikettendesign aus der .NLBL-Datei entnommen, und DEVMODE wendet die spezifische, druckerbezogene Formatierung an. Auf diese Weise ist es möglich, nur ein Etiketten-Master zu verwenden. Der Ausdruck wird auf jedem Drucker identisch sein, da jeweils die optimalen Druckereinstellungen angewandt werden.

Um den benutzerdefinierten DEVMODE auf ein Etikett anzuwenden, haben Sie zwei Optionen:

1. Anhand der Aktion Druckparameter festlegen, 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](#).

## 10.21 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 für den Dienst gelten. 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 den Automation Builder regelmäßig mit den Anmeldedaten des anderen Benutzerkontos ausführen möchten, nutzen Sie das in Windows verfügbare Befehlszeilen-Dienstprogramm **RUNAS**. Verwenden Sie die Schalter `/user`, um das Benutzerkonto anzugeben, und `/savecred`, damit Sie das Passwort nur einmal eingeben müssen; daraufhin wird es für die nächste Anmeldung gespeichert.

# 11 Beispiele

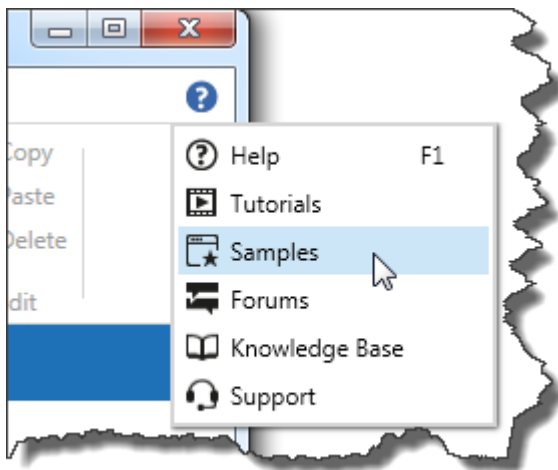
## 11.1 Beispiele

NiceLabel Automation wird mit Beispielen geliefert, die die Konfigurationsverfahren für häufig genutzte Datenstrukturen beschreiben und zur Konfiguration von Aktionen dienen. Auf diese Weise lernen Sie schnell, wie Sie Filter für die Extraktion von Daten aus Quellen wie CSV-Dateien (Dateien mit durch Kommas getrennten Werten), Datenexporten in veralteten Formaten, Druckerdateien, XML-Dokumenten und Binärdateien konfigurieren können.

Ein Link zum Ordner mit Beispielen findet sich in Automation Builder.

So öffnen Sie den Ordner mit Beispielen im Windows Explorer:

1. Öffnen Sie Automation Builder.
2. Klicken Sie auf das Fragezeichen in der oberen rechten Ecke.
3. Wählen Sie **Beispiele**.



4. Der Ordner mit den Beispieldateien wird im Windows Explorer geöffnet.
5. Siehe die Datei **README.PDF** in jedem Ordner.

Die Beispieldateien sind im folgenden Ordner installiert:

**BEISPIEL:** %PUBLIC%\Documents\NiceLabel 2017\Automation\Samples

Dies entspricht:

**c:\Users\Public\Documents\NiceLabel 2017\Automation\Samples**

# 12 Technischer Support

## 12.1 Online-Support

Sie finden die neuesten Versionen, Updates, Lösungen für Probleme und häufig gestellten Fragen (FAQs) auf der Produktwebsite unter [www.nicelabel.com](http://www.nicelabel.com).

Weitere Informationen finden Sie hier:

- Knowledgebase: <http://www.nicelabel.de/support/knowledge-base>
- NiceLabel Support: <http://www.nicelabel.de/support/technical-support>
- NiceLabel Tutorials: <http://www.nicelabel.de/learning-center/tutorials>
- NiceLabel Foren: <http://forums.nicelabel.com/>

**HINWEIS:** Wenn Sie einen Service-Vertrag (Service Maintenance Agreement, SMA) haben, wenden Sie sich bitte an den Premium-Support, wie im Vertrag angegeben.



Nord-, Mittel- und Südamerika  
+1 262 784 2456  
sales.americas@nicelabel.com

EMEA  
+386 4280 5000  
sales@nicelabel.com

Deutschland  
+49 6104 68 99 80  
sales@nicelabel.de

China  
+86 21 6249 0371  
sales@nicelabel.cn

[www.nicelabel.com](http://www.nicelabel.com)

