Loftware NiceLabel

Guide utilisateur NiceLabel Automation 10

Rev-2022-10

Table des matières

1. Bienvenue dans NiceLabel Automation 7
1.1. Architecture 8
1.2. Spécifications système requises 9
1.3. Installation 9
1.4. Activation 10
1.5. Mode d'essai 11
1.6. Onglet Fichier 11
1.6.1. Ouvrir 11
1.6.2. Compatibilité avec les produits NiceWatch 12
1.6.3. Enregistrer 14
1.6.4. Enregistrer sous 14
1.6.5. Options 15
1.6.6. À propos 23
1.6.6.1. Perdre votre connexion Label Cloud 24
2. Comprendre les filtres 25
2.1. Filtre de texte structuré 27
2.1.1. Filtre de texte structuré 27
2.1.2. Définition des champs 28
2.1.3. Activer la structure dynamique 31
2.2. Filtre de données non structurées 34
2.2.1. Filtre de données non structurées 34
2.2.2. Définition des champs 36
2.2.3. Définition de sous-zones 40
2.2.4. Définition des zones d'affectation 42
2.3. Configuration du filtre XML 44
2.3.1. Filtre XML 44
2.3.2. Définition des champs XML 45
2.3.3. Définir les éléments répétables dans le Filtre XML 48
2.3.4. Définition de la zone d'affectation XML 51
2.4. Configuration du filtre JSON 54
2.4.1. Filtre JSON 54
2.4.2. Définition des champs JSON 56
2.4.3. Définir les éléments répétables dans le Filtre JSON 59
2.4.4. Définition de la zone d'affectation JSON 64
2.5. Paramétrer le nom de l'étiquette et de l'imprimante dans les données entrées 67
3. Comprendre les déclencheurs 69
3.1. Déclencheur de Fichier 72
3.2. Déclencheur sur port série 76
3.3. Déclencheur de base de données 79
3.4. Déclencheur serveur TCP/IP 88
3.5. Déclencheur client TCP/IP 93
3.6. Déclencheur serveur HTTP 97
3.7. Déclencheur Web Service 103

3.8. Déclencheur de Cloud 114
3.8.1. Déployer le déclencheur de Cloud avec NiceLabel Cloud 115
3.8.1.1. Étapes du déploiement pour NiceLabel Cloud 115
3.8.1.2. Configurer le déclencheur de Cloud dans Automation Builder 116
3.8.1.3. Paramétrer l'accès au déclencheur de Cloud pour l'intégrateur externe 118
3.8.1.4. Créer une souscription sur le Portail développeur 119
3.8.1.5. Appeler votre déclencheur de Cloud (déploiement de NiceLabel Cloud) 122
3.8.1.6. Vérifier rapidement si votre déclencheur de Cloud fonctionne 123
3.8.2. Déployer le déclencheur de Cloud avec votre Control Center sur site 124
3.8.2.1. Configurer le déclencheur de Cloud dans Automation Builder 124
Général languette 124
Autre languette 126
Sécurité languette 127
Gâchette de démarrage 127
3.8.2.2. Appeler le déclencheur de Cloud (déploiement sur site) 128
3.9. Déclencheur de planification 128
3.9.1. Général 129
3.9.2. Récurrences 130
3.10. Test des déclencheurs 132
3.11. Utilisation de la couche de transport sécurisée (HTTPS) 135
3.12. Protéger la configuration du déclencheur de toute modification 139
3.13. Configurer un pare-feu pour des déclencheurs réseau 139
4. Variables 141
4.1. Utiliser des valeurs composées 142
4.2. Variables internes 143
4.3. Variables globales 145
5. Actions 147
5.1. Définition des actions 147
5.2. Actions indentées 147
5.3. Exécution d'Action 148
5.4. Actions conditionnelles 148
5.5. Identification des actions en état d'erreur de configuration 149
5.6. Désactiver les actions 149
5.7. Copier les actions 149
5.8. Naviguer dans la liste d'actions 150
5.9. Description des actions 150
5.10. Général 150
5.10.1. Ouvrir une étiquette 150
5.10.2. Imprimer l'étiquette 152
5.10.3. Exécuter le fichier de commande Oracle XML 156
5.10.4. Exécuter le fichier de commande SAP AII XML 158
5.10.5. Exécuter le fichier de commande 161
5.10.6. Envoyer des commandes personnalisées 163
5.11. Imprimante 164
5.11.1. Définir l'imprimante 164
5.11.2. Définir le nom du travail d'impression 166

5.11.3. Rediriger l'impression vers un fichier 168
5.11.4. Définir les paramètres d'impression 170
5.11.5. Rediriger l'impression vers un PDF 177
5.11.6. État de l'imprimante 180
5.11.7. Stocker l'étiquette sur l'imprimante 185
5.11.8. Imprimer un document PDF 187
5.12. Variables 189
5.12.1. Définir une variable 189
5.12.2. Enregistrer les données variables 190
5.12.3. Charger les données variables 192
5.12.4. Manipulation de chaîne de caractères 194
5.13. Impression par lot 197
5.13.1. Boucler 197
5.13.2. Utiliser un filtre de données 199
5.13.3. Pour chaque enregistrement 203
5.14. Données et connectivité 205
5.14.1. Ouvrir un Document / Programme 205
5.14.2. Enregistrer les données dans un fichier 208
5.14.3. Lecture des données d'un fichier 209
5.14.4. Effacer un fichier 212
5.14.5. Exécuter une requête SQL 213
5.14.6. Envoyer les données au port TCP/IP 219
5.14.7. Envoyer les données au port série 221
5.14.8. Lecture des données sur le port série 224
5.14.9. Envoyer les données à l'imprimante 226
5.14.10. Requête HTTP 228
5.14.11. Service Web 231
5.15. Autre 234
5.15.1. Récupérer les informations de l'étiquette 234
5.15.2. Exécuter un script 240
5.15.2.1. Éditeur de script 242
5.15.3. Message 243
5.15.4. Vérifier la licence 246
5.15.5. Essayer 247
5.15.6. Transformation XML 250
5.15.7. Grouper 253
5.15.8. Consigner les événements 254
5.15.9. Aperçu de l'étiquette 256
5.15.10. Créer une variante d'étiquette 258
6. Exécuter et gérer les déclencheurs 261
6.1. Déployer la configuration 261
6.2. Options de journalisation des événements 262
6.3. Gestion des déclencheurs 263
6.4. Utilisation du journal d'événements 265
6.5. Si votre configuration ne parvient pas à se charger... 268
7. Performances et options de retour d'informations 272

7.1. Traitement parallèle 272
7.2. Mise en cache de fichiers 273
7.3. Traitement d'erreur 276
7.4. Mode d'impression synchrone 277
7.4.1. Mode d'impression asynchrone 277
7.4.2. Mode d'impression synchrone 278
7.5. Retour d'informations sur le travail d'impression 280
7.6. Exclure les imprimantes de l'impression automatisée 283
7.7. Utiliser le mode d'impression Stocker/Rappeler 284
7.8. Cluster haute disponibilité (Fail-over) 285
7.9. Cluster de répartition des charges 286
8. Comprendre les structures de données 288
8.1. Fichiers binaires 288
8.1.1. Exemple 288
8.2. Fichiers de commande 289
8.2.1. Exemple 289
8.3. CSV composé 290
8.3.1. Exemple 290
8.4. Données existantes 290
8.4.1. Exemple 290
8.5. Base de données texte 291
8.5.1. Exemple 291
8.6. Données XML 292
8.6.1. Exemples 292
8.7. Données JSON 294
9. Référence et résolution de problèmes 298
9.1. Types de fichiers de commande 298
9.1.1. Caractéristiques des fichier de commande 298
9.1.2. Fichier de commande CSV 298
9.1.2.1. Exemple de fichier de Commande CSV 298
9.1.3. Fichier de commande JOB 299
9.1.3.1. Exemple de fichier de Commande JOB 299
9.1.4. Fichier de commande XML 300
9.1.4.1. Exemple de fichier de commande XML 300
9.1.5. Caractéristiques Oracle XML 305
9.1.5.1. XML DTD 305
9.1.5.2. Exemple d'XML Oracle 306
9.1.6. Caractéristiques SAP AII XML 307
9.1.6.1. Exemple SAP AII XML 307
9.1.7. Fichier de commande PAS 308
9.1.7.1. Exemple de fichier de commande PAS 308
9.2. Commandes personnalisées 308
9.2.1. Utiliser des commandes personnalisées 308
9.3. Accès aux ressources réseau partagées 316
9.4. Stockage de documents et contrôle des versions des fichiers de configuration 317
9.5. Accéder aux bases de données 318

9.5.1. Windows 32 bits 318
9.5.2. Windows 64 bits 318
9.6. Remplacement automatique de la police 318
9.7. Automatisation des rapports 321
9.7.1. Création de bases de données temporaires 322
9.7.2. Élaboration de rapports automatisés 323
9.7.3. Création de filtres de données 324
9.7.4. Création de déclencheurs pour votre nouveau filtre de données 324
9.8. Changer les paramètres d'impression multi threads par défaut 326
9.9. Compatibilité avec les produits NiceWatch 327
9.10. Contrôler le Service Automation avec les paramètres de ligne de commande 328
9.11. Remplacement de la chaîne de connexion à la base de données 332
9.12. Introduire des caractères spéciaux (Codes de Contrôle) 333
9.13. Liste des codes de contrôle 334
9.14. Attribution des licences et imprimantes utilisées 335
9.15. Fonctionnement en mode service 336
9.16. Ordre de recherche des fichiers 339
9.17. Sécuriser l'accès aux déclencheurs 339
9.18. Session d'impression 341
9.19. Conseils et astuces d'utilisation de variables dans les actions 343
9.20. Mode de Traçage 343
9.21. Comprendre les paramètres d'imprimante et DEVMODE 345
9.22. Utiliser le même compte utilisateur pour configurer et exécuter les déclencheurs 347

1. Bienvenue dans NiceLabel Automation

NiceLabel Automation est une application d'automatisation des actions répétitives. En général, elle est utilisée pour intégrer un processus d'impression d'étiquettes dans un système d'informations existant : soit une application de bureau existante, soit des lignes de production et d'emballage, soit des systèmes de distribution et chaînes d'approvisionnement. Avec NiceLabel Automation, les applications utilisées dans toutes les divisions et sur tous les sites d'une entreprise peuvent imprimer des étiquettes en utilisant des masques d'étiquettes approuvés.

NiceLabel Automation vous aide à déployer et exécuter un système d'impression d'étiquettes optimal en synchronisant les événements de l'activité commerciale avec la production d'étiquettes. L'impression automatique sans interaction humaine est de loin la manière la plus efficace d'éliminer les erreurs humaines et maximiser les performances.

L'automatisation de l'impression d'étiquettes via une application basée sur des déclencheurs tourne autour des trois processus principaux suivants :

Déclencheur
Les déclencheurs sont une fonction simple mais puissante qui permettent d'automatiser votre travail. Le déclencheur est une instruction de cause à effet: «Quand un événement surveillé survient, il effectue une action.»

Il s'agit du traitement **IF .. THEN** (si .. alors). Les déclencheurs savent bien gérer les événements répétitifs.

L'impression automatique d'étiquettes est déclenchée par une opération dans l'entreprise.NiceLabel Automation est paramétré pour surveiller un dossier, un fichier, une communication ou un port. Si une opération s'effectue dans l'entreprise, un changement de fichier ou des données entrantes sont détectés. le processus d'impression d'étiquettes est activé.

En savoir plus sur les différents Déclencheurs :

- Déclencheur de fichier
- Déclencheur sur port série
- Déclencheur de base de données
- Déclencheur de planification
- Déclencheur serveur TCP/IP
- Déclencheur client TCP/IP
- Déclencheur HTTP
- Déclencheur Web Service
- Déclencheur de Cloud

Extraction de données et placement
Lorsqu'un déclencheur lance l'impression, NiceLabel Automation extrait les données de l'étiquette et les insère dans les objets variables placés sur une étiquette.

Les Filtres d'extraction de données peuvent utiliser :

- Des fichiers de Texte Structurés
- Des fichiers de Texte non Structurés
- Différents fichiers XML
- Fichiers JSON

Exécution d'action
Après avoir fait correspondre les données avec les objets variables sur une étiquette, NiceLabel Automation commence à exécuter des actions. Un jeu d'actions basique inclut généralement **Ouvrir l'étiquette** et **Imprimer l'étiquette** pour imprimer les données extraites sur une étiquette. Les données peuvent aussi être envoyées à des emplacements de fichier en local ou réseau, à un serveur Web, à des périphériques, etc.

Il y a plus de 30 actions disponibles. Ensemble, elles englobent un large éventail de scénarios usuels dans les environnements d'entreprise actuels.

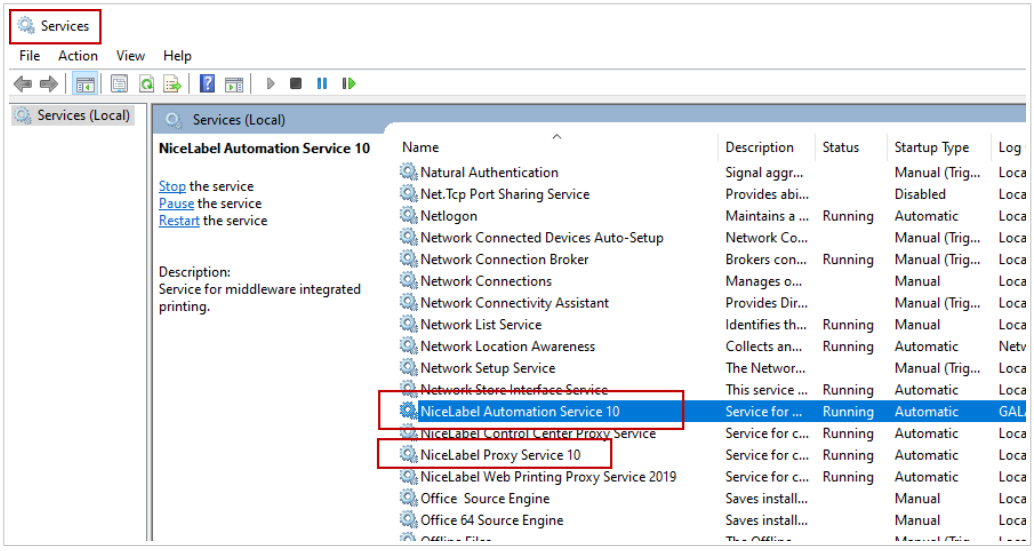
Plus d'informations concernant les Actions d'impression de base et avancée.

1.1. Architecture

NiceLabel Automation est une application basée sur un service. L'exécution de toutes les règles et actions s'effectue en arrière-plan sous les infos d'identification du compte d'utilisateur défini pour le Service.

NiceLabel Automation se compose de trois choses.

- Automation Builder**:** Le développeur utilise cette application pour créer des déclencheurs, des filtres et des actions, et les utilise dans une configuration de travail. Les actions qui appartiennent à une telle configuration s'exécutent lorsqu'un déclencheur a reçu des données. Cette application s'exécute toujours en tant qu'application 32 bits.
- Automation Manager**:** Il s'agit de l'application de gestion qui surveille l'exécution des déclencheurs en temps réel et leur ordonne de démarrer ou de s'arrêter.Automation Manager s'exécute toujours en tant qu'application 32 bits.
- NiceLabel Automation**Service:** C'est le moteur d'impression qui exécute les règles définies dans les déclencheurs. Il y a deux applications de service : le Service NiceLabel Automation et le Service Proxy NiceLabel. Le Service détecte toujours le nombre de bits de la machine Windows et fonctionne au même niveau (par ex. en application 64 bits sous Windows 64 bits), alors que le Service Proxy sera toujours en 32 bits.



1.2. Spécifications système requises

NOTE

Toujours bien vérifier les dernières spécifications système requises sur cette page web : Spécifications système requises.

1.3. Installation

NOTE

Ci-dessous une description sommaire de la procédure d'installation. Pour plus d'informations, consultez le NiceLabel Automation Guide d'installation.

Avant de commencer l'installation, il faut que l'infrastructure soit compatible avec Spécifications système requises.

Pour installer NiceLabel Automation, effectuer les opérations suivantes :

- Téléchargez le fichier d'installation depuis le NiceLabel site Internet, puis exécutez le fichier exécutable que vous avez téléchargé.
- Insérer le DVD NiceLabel.
    1. Le menu principal de l'application démarre automatiquement.
       Si le menu principal de l'application ne démarre pas, double-cliquer sur le fichier **START.EXE** du DVD.
    2. Cliquer sur **Installer** NiceLabel.

3. Suivre les instructions de l'**Assistant d'installation**.
   Durant l'installation, l'interface vous propose de saisir le nom de l'utilisateur sous lequel le service NiceLabel Automation va fonctionner. Assurez-vous de sélectionner un vrai nom d'utilisateur, car le service hérite des droits associés à ce nom d'utilisateur spécifique. Pour plus d'informations, consulter l'article Fonctionnement en mode service .

Mise à jour de version

Pour effectuer la mise à jour de la dernière mouture de NiceLabel Automation, installer la nouvelle version au-dessus de l'autre qui sera écrasée. L'ancienne version sera supprimée pendant la mise à jour et remplacée par la nouvelle. Les paramètres existants seront conservés. La mise à jour supprime le contenu de la base de données du journal.

1.4. Activation

Activer NiceLabel Automation pour activer le traitement des déclencheurs configurés. La procédure d'activation requiert une connexion Internet, si possible sur la même machine qui exécute le logiciel. Utiliser la même procédure d'activation pour activer la clé de licence d'essai.

> **NOTE**
> Vous pouvez activer le logiciel à partir de Automation Builder ou de Automation Manager.

Activation dans Automation Builder

1. Exécuter Automation Builder.
2. Sélectionner **Fichier > À propos > Activer la licence**.
   L'Assistant d'installation se lance.
3. Suivre les instructions à l'écran.

Activation dans Automation Manager

1. Exécuter Automation Manager.
2. Aller à l'onglet **À propos**.
3. Cliquer sur **Activer la licence**.
4. Suivre les instructions à l'écran.

1.5. Mode d'essai

Le mode d'essai vous permet de tester le produit NiceLabel Automation durant une période de 30 jours. Le mode d'essai propose les mêmes fonctionnalités qu'une licence, vous permettant ainsi d'évaluer le produit avant de l'acheter.Automation Manager affiche la notification d'essai en permanence ainsi que le nombre de jours restants pour l'essai. Lorsque le mode d'essai expire, le service NiceLabel Automation arrête de traiter les déclencheurs. Le décompte des 30 jours commence le jour de l'installation.

NOTE

Pour étendre la période d'essai, contacter votre revendeur NiceLabel pour lui demander une nouvelle clé d'essai. Il faut activer la clé de licence d'essai. Pour plus d'informations, voir la section Activation.

1.6. Onglet Fichier

L'onglet **Fichier** sert de panneau de gestion de documents. Liste des options disponibles :

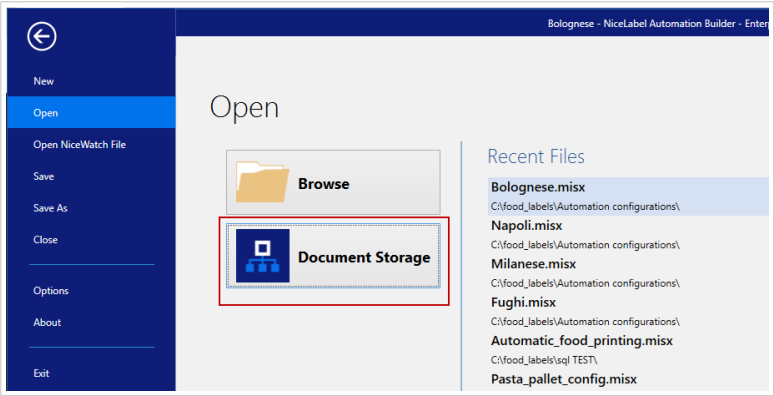
- **Nouveau :** pour créer un nouveau fichier de configuration.
- Ouvrir : pour ouvrir les fichiers de configuration existants.
- **Ouvrir un fichier NiceWatch :** ouvre une ancienne configuration NiceWatch de NiceLabel.
- Enregistrer : enregistre le fichier de configuration actif.
- Enregistrer sous : permet d'enregistrer le fichier de configuration actif en donnant son nom et son emplacement.
- Options : ouvre la boite de dialogue pour configurer les programmes par défaut.
- À propos : fournit les informations sur la licence et la version du logiciel.
- **Quitter**: ferme l'application.

1.6.1. Ouvrir

La boîte de dialogue **Ouvrir** permet d'ouvrir les fichiers de configuration existants dans Automation Builder.

Parcourir permet de sélectionner le fichier de configuration sur un lecteur local ou connecté en réseau.

Stockage de documents ouvre le stockage de documents du NiceLabel Control Center connecté. L'onglet **Stockage de documents** permet de Gérer la copie du fichier de configuration enregistré.



Le champ **Fichiers récents** liste les derniers fichiers de configuration modifiés. Cliquer sur l'un d'entre eux pour ouvrir le fichier.

1.6.2. Compatibilité avec les produits NiceWatch

NiceLabel Automation vous permet de charger des configurations de déclencheurs conçues en utilisant les produits NiceWatch. Dans la majorité des cas, une configuration de NiceWatch peut être exécutée dans NiceLabel Automation sans aucune modification.

Les produits NiceLabel Automation utilisent les nouveaux moteurs d'impression en .NET, optimisés en performances avec un faible encombrement mémoire. Le nouveau moteur d'impression ne supporte pas toutes les options de création des étiquettes disponibles dans l'éditeur d'étiquettes. Chaque nouvelle mouture de NiceLabel Automation en diminue le nombre mais certaines fonctionnalités restent indisponibles.

Résoudre les problèmes d'incompatibilité

NiceLabel Automation émet une alerte à chaque tentative d'impression d'étiquettes existantes qui contiennent des fonctionnalités indisponibles dans le nouveau moteur d'impression.

S'il y a des incompatibilités entre les fichiers de configuration ou les masques d'étiquettes NiceWatch, Automation vous avertit concernant :

- **Compatibilité avec la configuration de déclencheur :** À l'ouverture de la configuration de NiceWatch (fichier .MIS), NiceLabel Automation la compare aux éléments supportés. Tous les fonctionnalités de NiceWatch ne sont pas disponibles dans NiceLabel Automation. Certaines sont totalement indisponibles, d'autres sont configurées différemment. Si le fichier .MIS contient des fonctionnalités non supportés, elles seront listées. Automation supprime ces fonctionnalités de la configuration.
  Dans ce cas, il faudra ouvrir le fichier .MIS dans Automation Builder et résoudre les problèmes d'incompatibilité. Il faudra utiliser la fonctionnalité disponible de NiceLabel Automation pour recréer la configuration.
- **Compatibilité avec les masques d'étiquettes :** Si vos masques d'étiquettes existants contiennent des fonctionnalités de moteur d'impression non supportées telles que fournies par NiceLabel Automation, vous verrez des messages d'erreurs dans le panneau **Journal**. Cette information est visible dans Automation Builder (durant la création des déclencheurs) ou dans Automation Manager (lors de l'exécution des déclencheurs).
  Dans ce cas, il faut ouvrir le fichier de l'étiquette dans l'éditeur d'étiquettes et enlever de l'étiquette les éléments non supportés.

> **NOTE**
> Pour plus d'informations concernant les incompatibilités avec NiceWatch et l'éditeur d'étiquettes, consulter laNiceLabel Automation Compatibility with NiceLabel Designers V6 and NiceWatch V5.

Ouvrir la configuration NiceWatch pour l'éditer

Ouvrir la configuration NiceWatch existante (fichier .MIS) dans Automation Builder et l'éditer dans Automation Builder. Vous pouvez enregistrer la configuration en tant que fichier .MISX uniquement.

Pour éditer la configuration NiceWatch, effectuer les opérations suivantes :

1. Démarrer Automation Builder.
2. Sélectionner **Fichier** > **Ouvrir fichier NiceWatch**.
3. Dans la boîte de dialogue **Ouvrir**, rechercher le fichier de configuration NiceWatch (fichier .MIS).
4. Cliquer sur **OK**.
5. Si la configuration contient des fonctionnalités non supportées, elles sont affichées sous forme de liste. Automation les supprime de la configuration.

Ouvrir la configuration NiceWatch pour l'exécuter

Il est possible d'ouvrir la configuration NiceWatch (fichier .MIS) dans Automation Manager sans conversion au format de fichier NiceLabel Automation (fichier .MISX). Si les déclencheurs de NiceWatch sont compatibles avec NiceLabel Automation, ils sont directement utilisables.

Pour éditer et déployer la configuration NiceWatch, effectuer les opérations suivantes :

1. Démarrer Automation Manager.
2. Cliquer sur le bouton **+ Ajouter**.
3. Dans la boîte de dialogue **Ouvrir**, changer le type de fichier en **Configuration NiceWatch**.
4. Rechercher le fichier de configuration NiceWatch (fichier .MIS).
5. Cliquer sur **OK**.
6. Le déclencheur de la configuration sélectionnée sera affiché dans Automation Manager. Pour lancer le déclencheur, le sélectionner et cliquer sur **Démarrer**.

NOTE

S'il y a un problème de compatibilité avec la configuration NiceWatch, il faudra l'ouvrir dans Automation Builder et le reconfigurer.

1.6.3. Enregistrer

Enregistrer enregistre le fichier de configuration actif en utilisant le nom qu'elle avait à l'ouverture.

NOTE

Quand la configuration est ouverte pour la première fois, **Enregistrer** dirige directement vers la boîte de dialogue **Enregistrer sous**.

1.6.4. Enregistrer sous

Enregistrer sous permet d'enregistrer le fichier de configuration actif en donnant son nom et son emplacement. Vous pouvez enregistrer vos fichiers de configuration vers votre disque local ou vers **Stockage de documents** dans votre NiceLabel Control Center

Dossiers récents ce champ liste les dossiers récemment utilisés pour enregistrer les fichiers de configuration.

1.6.5. Options

Utiliser les paramètres de cette boîte de dialogue pour personnaliser l'application. Sélectionner un groupe à partir des panneaux de gauche et configurer ses paramètres.

Dossiers

Ce panneau vous permet de sélectionner les dossiers de stockage par défaut pour les étiquettes, les formulaires, les bases de données et les fichiers d'images. Par défaut, l'emplacement du dossier est dans les Documents de l'utilisateur. Il y aussi les dossiers par défaut dans lesquels NiceLabel Automation recherche les fichiers dont vous donnez le nom sans le chemin entier. Pour plus d'informations sur l'ordre de recherche de fichier, voir l'article Ordre de recherche pour les fichiers requis.

Les changements relatifs aux dossiers se propagent au Service NiceLabel Automation en moins d'une minute. Pour appliquer les modifications immédiatement, redémarrer le service.

NOTE

Les paramètres appliqués ici sont enregistrés dans le profil de l'utilisateur connecté. Si le Service NiceLabel Automation fonctionne sous un autre compte utilisateur, il faut se connecter à Windows en utilisant cet autre compte, et changer le dossier d'étiquettes par défaut. Vous pouvez aussi utiliser l'utilitaire Windows en ligne de commande RUNAS pour lancer Automation Builder sous cet autre utilisateur.

Langue

Le panneau Langue permet de choisir la langue de l'interface de NiceLabel Automation. Sélectionner la langue désirée et cliquer sur **OK**.

NOTE

Le changement s'applique au redémarrage de l'application.

Variables globales

Le panneau Variables globales définit l'emplacement à utiliser pour stocker les variables globales :

- **Utiliser les Variables globales stockées sur le serveur (NiceLabel Control Center) :** Définit l'emplacement de stockage des variables globales sur NiceLabel Control Center.

  > **INFO NIVEAU DE PRODUIT**
  > Cette option devient disponible avec LMS Pro ou les licences NiceLabel LMS Enterprise.

- **Utiliser les variables globales stockées dans un fichier (local ou partagé) :** Définit l'emplacement de stockage des variables globales dans un dossier en local ou partagé. Saisir le chemin exact ou cliquer sur **Ouvrir** pour localiser le fichier.

Imprimantes sous licence

> **NOTE**
> Conserver les informations sur l'utilisation des imprimantes sous licence est possible avec les licences multi-utilisateurs.

Le panneau **Imprimantes sous licence** fournit les informations journalisées concernant les imprimantes utilisées pour l'impression.

Le groupe **Informations sur les imprimantes sous licence** affiche le nombre d'imprimantes autorisé utilisées après avoir imprimé sur plusieurs imprimantes.

- **Nombre d'imprimantes autorisé par licence :** Nombre d'imprimantes utilisables avec la licence NiceLabel 10 en cours.
- **Nombre d'imprimantes utilisées au cours des 7 derniers jours :** Nombre d'imprimantes utilisées par NiceLabel 10 au cours des 7 derniers jours.

  > **ASTUCE**
  > Pendant une période de 7 jours, la licence NiceLabel 10 permet d'utiliser uniquement le nombre spécifié d'imprimantes différentes.

> **AVERTISSEMENT**
> La licence achetée définit le nombre d'imprimantes autorisées. Après avoir dépassé ce nombre, un avertissement apparaît. Après avoir doublé le nombre d'imprimantes autorisé, vous ne pouvez plus imprimer sur d'autres imprimantes.

Voir les états de l'impression dans plusieurs colonnes :

- **Imprimante :** Nom ou modèle de l'imprimante sélectionnée pour l'impression.

  > **NOTE**
  > Si l'imprimante connectée est partagée, seul le modèle d'imprimante s'affiche.

- **Emplacement :** Nom de l'ordinateur qui a émis le travail d'impression.
- **Port :** Port utilisé par l'imprimante.
- **Dernière utilisation :** Temps écoulé depuis le dernier travail d'impression.
- **Réservée :** Évite que l'imprimante soit supprimée après une veille de plus de 7 jours.

> **NOTE**
> Quand une imprimante est inutilisée pendant plus de 7 jours, elle est supprimée automatiquement à moins que l'option **Réservée** soit cochée.

Le panneau **Permissions** permet de verrouiller l'utilisation de l'imprimante sur un poste de travail.

- **Ce poste de travail peut seulement utiliser les imprimantes réservées :** Quand cette option est activée, seules les imprimantes réservées sont autorisées pour éditer et imprimer les étiquettes dans NiceLabel 10.

  > **ASTUCE**
  > Utiliser cette option pour éviter de dépasser le nombre d'imprimantes autorisé par la licence en imprimant sur des imprimantes indésirables ou des applications d'impression dans un fichier. Réserver les imprimantes d'étiquettes thermiques ou laser et limiter l'impression à ce type d'imprimantes uniquement. Ainsi, l'impression d'étiquettes est continue, avec une licence multi-utilisateurs.

  Cette option peut également être activée en utilisant le fichier `product.config` :

  1. Rechercher le dossier système.
     %PROGRAMDATA%\NiceLabel\NiceLabel 10
  2. Faire une copie de sauvegarde du fichier **`product.config`**.
  3. Ouvrir **`product.config`** dans un éditeur de texte. Le fichier a une structure XML.
  4. Ajouter les lignes suivantes :

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

  5. Enregistrer le fichier. Le `exemple d'imprimante`est réservé.

Control Center

Le panneau **Control Center** permet d'activer et configurer le contrôle des événements et des travaux d'impression. NiceLabel Control Center active les rapports centralisés des événements et des travaux d'impressions, et le stockage centralisé des variables globales.

> **NOTE**
> Cet onglet devient disponible avec les licences LMS Pro ou LMS Enterprise.

Le groupe **Adresse** permet de définir le serveur NiceLabel Control Center à utiliser.

- **Adresse du serveur de Control Center :** URL du serveur NiceLabel Control Center connecté. Le sélectionner dans la liste déroulante automatique des serveurs trouvés sur le réseau ou saisir à la main l'adresse d'un serveur.

> **NOTE**
> Les clés des licences du serveur NiceLabel Control Center et des postes de travail doivent correspondre pour permettre la connexion.

Le groupe **Gestion des événements** définit les types d'événements à inscrire dans le journal NiceLabel Control Center.

- **Evènements d'impression :** Enregistre les événements liés à l'impression à partir des postes de travail.
- **Survenance d'erreur :** Enregistre toutes les erreurs rapportées.

> **NOTE**
> Par défaut, toutes les impressions et les erreurs sont inscrites dans le journal par NiceLabel Control Center.

- **Activité du déclencheur :** Enregistre tous les déclencheurs activés.
- **Modification de l'état du déclencheur :** Enregistre les changements d'état du déclencheur dus à des déclencheurs précédents.

Le groupe **Contrôle des impressions** permet d'inscrire dans le journal toutes les impressions terminées ou en cours dans NiceLabel Control Center.

- **Enregistrer le journal d'impression sur le serveur :** Active le journal des travaux d'impression.
- **Contrôle détaillé de l'impression :** Active le contrôle des états renvoyés par l'imprimante connectée.

NOTE

Pour que cette option soit disponible, il faut que :

- L'imprimante soit bidirectionnelle.
- NiceLabel Que le pilote d'imprimante utilisé soit un pilote.

Automation

Ces paramètres définissent les fonctionnalités avancées de l'application.

> **NOTE**
> Les changements s'appliqueront après redémarrage de l'application.

Communication du service

- **Port de communication du service :** Automation Manager contrôle le service en utilisant le protocole TCP/IP sur le port sélectionné. Si le port par défaut ne vous convient pas pour l'utiliser sur l'ordinateur, sélectionner un autre numéro de port. Attention à ne pas sélectionner un numéro de port déjà utilisé par une autre application.

Journal

- **Effacer les entrées du journal tous les jours à :** Définit le début du processus de nettoyage. Les anciens événements sont effacés de la base de données du journal.
- **Effacer les entrées du journal lorsqu'elles ont plus de (jours) :** Spécifie la conservation de l'événement dans la base de données du journal. Tous les événements plus anciens que le nombre de jours spécifié sont effacés de la base de données lors du nettoyage de la base de données du journal.
- **Messages du journal :** Spécifie la mesure de l'enregistrement de l'événement dans le journal. Durant les phases de développement et de test, il vaut mieux activer une journalisation détaillée. Dans ce cas, activer tous les messages pour améliorer le traçage de l'exécution des déclencheurs. Plus tard, pendant la phase de production, réduire la quantité de messages pour activer seulement la journalisation des erreurs.

Performance

- > **INFO NIVEAU DE PRODUIT**
  > La fonctionnalité de cet élément est disponible dans LMS Enterprise.

  Fichiers en cache du Stockage de documents et du partage réseau : Pour améliorer le temps de sortie de la première étiquette et accroître les performances générales, NiceLabel Automation permet la mise en cache de fichiers. Après avoir chargé des étiquettes, images et données de bases de données provenant d'un réseau en partage, tous les fichiers requis doivent être récupérés avant que l'impression ne commence.

  > **ASTUCE**
  > Si les fichiers sont mis en cache localement, l'effet de lenteur du réseau est éliminé, car ils sont tous chargés depuis votre disque local.

Automation utilise le dossier local suivant comme cache pour les fichiers distants : `%PROGRAMDATA%\NiceLabel\NiceLabel 10\FileCache`.

- **Intervalle entre les mises à jour (minutes) :** Définit l'intervalle de temps en minutes durant lequel le cache est synchronisé avec les fichiers dans leur dossier d'origine. C'est l'intervalle de temps durant lequel le système est autorisé à utiliser une version qui n'est pas forcément la dernière.
- **Effacer les fichiers en cache non utilisés au bout de (jours) :** Définit l'intervalle de temps au bout duquel tous les fichiers sont effacés du cache.

> **NOTE**
> On peut mettre en cache des fichiers d'étiquettes ou d'images. Après avoir activé la mise en cache des fichiers, redémarrer le service Automation pour que la modification prenne effet.

- **Dossiers en précache du Stockage de documents** vous permet de mettre en cache localement les fichiers issus des dossiers du Stockage de documents de Control Center sur votre ordinateur. En autorisant la mise en précache, le contenu du cache local continue de se synchroniser avec les dossiers de stockage de documents sélectionnés.

> **NOTE**
> Par rapport à la mise en cache, la mise en précache réduit le temps d'impression pour votre première étiquette imprimée.

Ajouter chaque dossier de stockage de documents dans une ligne séparée.

```
/Labels/Folder1

/Labels/Folder2
```

> **NOTE**
> Votre Automation Builder doit être connecté à Control Center pour activer la synchronisation hors ligne des fichiers en cache.

Support d'ensemble (Cluster)

> **INFO NIVEAU DE PRODUIT**
> La fonctionnalité de cet élément est disponible dans LMS Enterprise.

Ce paramètre active la possibilité d'utiliser NiceLabel Automation en mode cluster haute disponibilité (failover). Sélectionner le dossier utilisé par les deux nœuds du cluster pour partager le statut des déclencheurs en temps réel.

Designer

Le panneau **Designer** permet de configurer le comportement de NiceLabel 10 à l'ouverture.

- **Afficher chaque document dans propre fenêtre :** Si cette option est activée, les documents additionnels ouverts apparaissent dans différentes fenêtres de NiceLabel 10. Ceci est valable tant pour les documents nouveaux que pour les documents existants.
  Si vous décidez de désactiver cette option, les documents additionnels ouverts apparaissent dans l'instance actuellement active de NiceLabel 10.
- La **source des paramètres de l'imprimante** vous permet de choisir la source des paramètres de l'imprimante.
  - **Utiliser les paramètres d'impression du pilote d'imprimante :** Sélectionner cette option si vous préférez imprimer en utilisant les paramètres du pilote d'imprimante. Cette option vous permet de normaliser les paramètres de l'imprimante dans votre environnement de travail.
  - **Utiliser les paramètres d'imprimante personnalisés enregistrés dans l'étiquette :** Chaque étiquette peut avoir ses propres paramètres d'imprimante définis et enregistrés par l'utilisateur. Sélectionner cette option pour utiliser ces paramètres personnalisés pour vos étiquettes à l'impression.

1.6.6. À propos

La boîte de dialogue À propos fournit des informations sur la licence du produit NiceLabel, permet d'acheter la licence (quand le produit est en mode essai) et de l'activer. Elle donne aussi des détails sur le logiciel et permet de changer de niveau de produit NiceLabel 10.

Le groupe **Informations sur la licence** comporte :

- **Durée du mode test :** Les informations sur le nombre de jours restants pour évaluer le produit. Ce segment n'est plus visible après achat et activation de la licence.
- **Acheter la licence :** Bouton d'accès direct au site de vente en ligne de NiceLabel.
- **Activer la licence :** Ce bouton ouvre la boîte de dialogue d'activation de la licence NiceLabel 10. Consulter le NiceLabel 10guide d'installation de pour plus de détails sur le processus d'activation de la licence. Après activation de la licence, ce bouton est renommé en **Désactiver la licence**. Après avoir cliqué dessus et confirmé la désactivation, le produit NiceLabel 10 n'est plus activé.
- **Changer le niveau du produit :** Ouvre la boîte de dialogue pour sélectionner le niveau de produit. En mode Test, il est possible de choisir et évaluer tous les niveaux de produits. Quand la licence est activée, il est possible de changer le niveau de produit mais seulement sur les niveaux inférieurs.

NOTE

Le changement de niveau de produit ne prend effet qu'après redémarrage de l'application.

- **Mettre à jour la licence :** Ouvre la boîte de dialogue de mise à jour du niveau de produit. Consulter le guide d'installation de NiceLabel 10 pour plus de détails sur le processus de mise à jour de la licence.

Le groupe **Informations sur le logiciel** comporte des détails sur la version du logiciel installé et le numéro de mouture.

1.6.6.1. Perdre votre connexion Label Cloud

Si votre Automation Manager est connecté à Label Cloud et que vous perdez la connexion à Internet, vous devez rétablir la connexion dans les cinq jours. Si vous ne vous reconnectez pas à Label Cloud, Automation Manager se ferme automatiquement.

Après avoir perdu la connexion à Internet, et si votre ordinateur reste hors ligne, un avertissement apparaît sous cinq jours. Automation Manager se ferme cinq minutes après que vous ayez vu cet avertissement.

Après avoir rétabli la connexion Internet, ouvrez Automation Builder ou Automation Manager et connectez-vous à Label Cloud. Votre copie sera à nouveau active.

AVERTISSEMENT

Enregistrez votre travail sur un emplacement hors ligne (votre ordinateur) pour ne pas perdre vos modifications.

2. Comprendre les filtres

NiceLabel Automation utilise des filtres pour définir la structure des données reçues par les déclencheurs. Chaque fois qu'un déclencheur reçoit des données, un ou plusieurs filtres analysent les données reçues. Ce processus extrait les valeurs pertinentes pour votre configuration. Chaque filtre inclut des règles pour identifier les champs dans les données reçues.

> **NOTE**
> Le résultat fourni par le filtre est une liste de champs et leurs valeurs (**`paires nom-valeur`**).

Types de filtres

Pour plus d'informations, consulter les sections Filtre de texte structuré, Filtre de données non structurées, Filtre XML, et Filtre JSON.

Structure de données

La complexité du filtre dépend de la structure des données. Les fichiers aux données structurées, tels que CSV ou XML, facilitent l'extraction. Dans ce cas, les noms des champs sont déjà définis par les données. L'extraction des paires **`nom-valeur`** est rapide. Quand les données n'ont pas de structure claire, la définition des règles d'extraction prend plus de temps. Vous pouvez tomber sur ce type de données en exportant des documents et des rapports d'anciens systèmes, de communication interceptée entre deux périphériques, et d'une capture d'un flux d'impression.

Le filtre définit une liste de champs qui sont extraits des données entrantes dès l'exécution du filtre.

NiceLabel Automation supporte différents types de données d'entrée qui peuvent tous être analysés par l'une des méthodes de filtrage. Assurez-vous de choisir le filtre qui correspond au type de données entrantes. Par exemple, le **Filtre de texte structuré** pour des données CSV entrantes, le **Filtre JSON** pour des données JSON entrantes, et le **Filtre XML** pour des données XML entrantes. Pour toutes données non structurées, utiliser le **Filtre de données non structurées**. Pour plus d'informations, voir l'article Comprendre les structures de données.

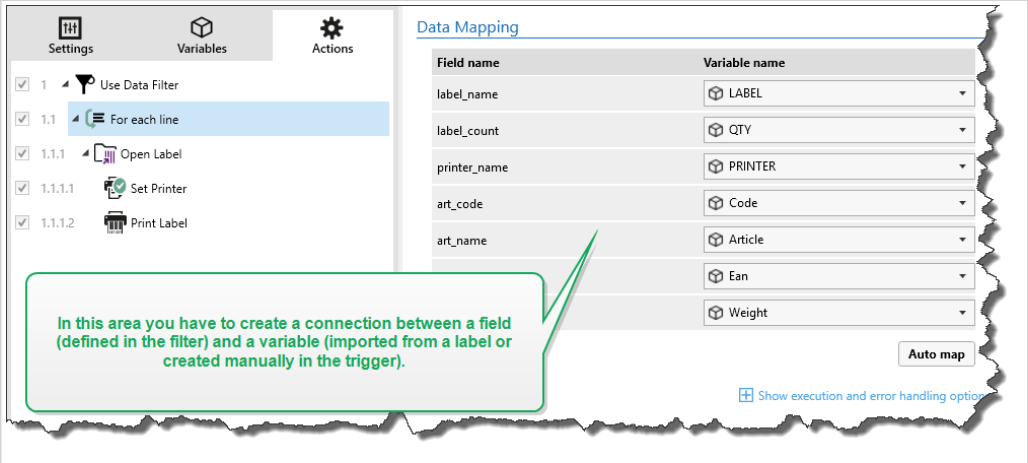
Extraction de données

Le filtre est un ensemble de règles qui n'effectue aucune extraction par lui-même. Pour exécuter le filtre, lancer l'action Utiliser le Filtre de Données. Cette action exécute les règles du filtre sur les valeurs et extrait les informations.

Chaque type de déclencheur peut exécuter autant d'actions Utiliser le Filtre de Données que nécessaires. Si vous recevez des données d'entrée composées qui ne peuvent pas être analysées par un seul filtre, définir et exécuter les règles de plusieurs filtres dans une séquence d'actions Utiliser le Filtre de Données. À la fin, utiliser les valeurs extraites par toutes les actions pour une même étiquette.

Associer les champs aux variables

Pour utiliser les valeurs extraites, il faut les enregistrer dans des variables. L'action Utiliser le Filtre de Données extrait les valeurs et les enregistre dans les variables. Pour configurer ce processus, mapper chaque variable au champ respectif. La valeur du champ est ensuite enregistrée dans la variable mappée.



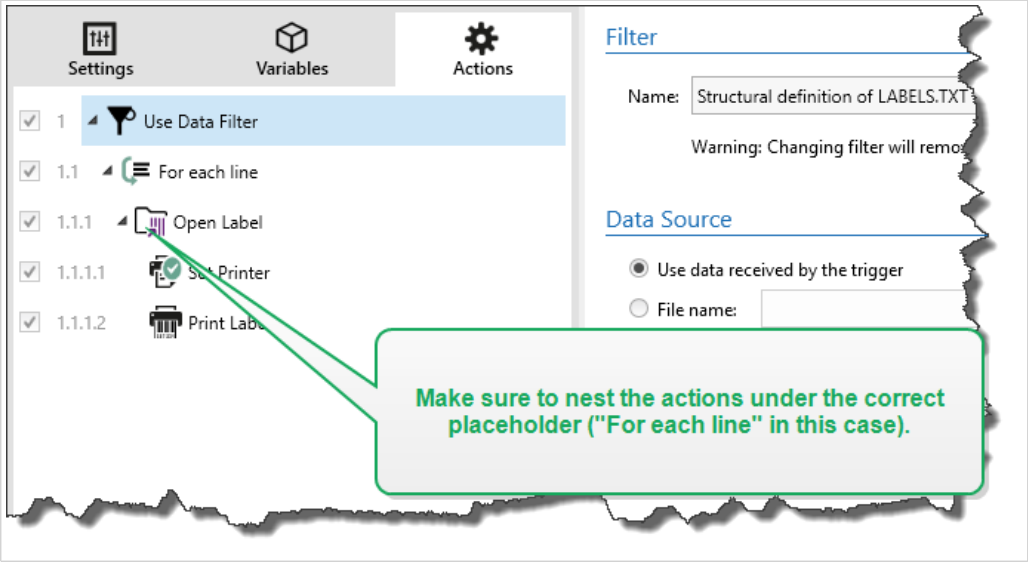
ASTUCE

Il est conseillé de définir les champs et variables dont les noms correspondent. Dans ce cas, le mappage automatique associe les variables aux champs de mêmes noms, éliminant le besoin de mappage manuel.

Le mappage automatique est disponible pour tous les types de filtres. Quand le mappage automatique est activé, l'action Utiliser le Filtre de Données extrait automatiquement les valeurs et les associe aux variables ayant le même nom que les champs. Pour plus d'informations, consulter les sections Activer la structure dynamique pour Filtre de texte structuré, Définir la zone d'affectation pour Filtre de données non structurées et Définir la zone d'affectation pour les filtres XML ou JSON.

Exécuter des actions avec l'extraction de données

Généralement, il faut exécuter des actions avec les données extraites, telles qu'**Ouvrir l'étiquette**, **Imprimer l'étiquette** ou une des actions de connectivité de sortie. Il est extrêmement important d'indenter ces actions dans l'action **Utiliser le Filtre de Données**. Ainsi, les actions indentées sont exécutées pour chaque extraction de données.



Exemple

Avec un fichier CSV de 5 lignes, l'action indentée est aussi exécutée 5 fois, une fois pour chaque extraction de données. Si les actions ne sont pas indentées, elles ne s'effectuent qu'une seule fois et contiennent les données de la dernière extraction de données. Comme pour l'exemple ci-dessus, la 5e ligne du CSV serait imprimée avec les quatre premières lignes non imprimées. Avec des sous-zones, il faut indenter votre action dans le bon espace réservé.

2.1. Filtre de texte structuré

2.1.1. Filtre de texte structuré

Pour en savoir plus sur les filtres en général, consulter l'article Comprendre les filtres.

Utiliser ce filtre pour chaque réception d'un fichier texte structuré. Ce sont des fichiers texte dans lesquels les champs sont identifiés par l'une des méthodes suivantes :

- **Les champs sont délimités par un caractère :** Les caractères de délimitation habituels sont la virgule ou le point-virgule. Le fichier CSV (valeurs séparées par une virgule) est un exemple typique.
- **Les champs contiennent un nombre fixe de caractères :** Autrement dit, les champs sont définis par des colonnes de largeur fixe.

Pour des exemples de données de texte structuré, voir l'article Base de données texte.

Définition de la structure

Voici les options suivantes permettant de définir la structure du fichier de texte :

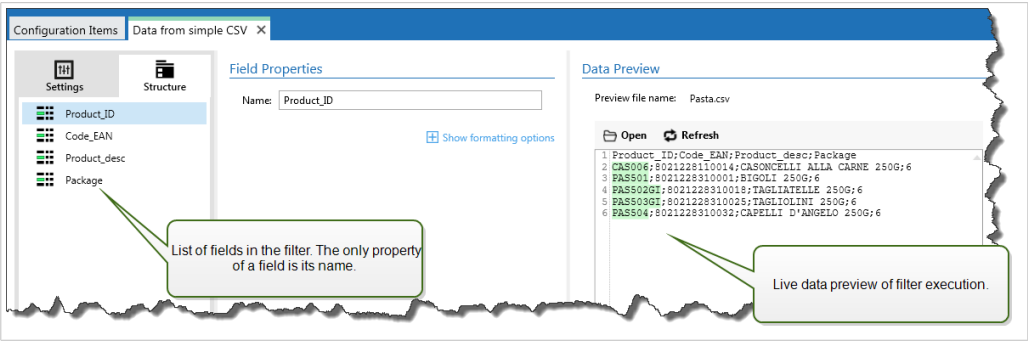
- **Importer la structure en utilisant l'Assistant de fichier texte :** Dans ce cas, cliquer sur le bouton **Importer la structure des données** dans le ruban et suivre les instructions à l'écran. Quand l'Assistant est terminé, le type de base de données texte et tous les champs sont définis. Si la première ligne de données contient les noms de champs, l'Assistant peut les importer. Il s'agit de la méthode recommandée si le déclencheur reçoit toujours les données dont la structure reste inchangée.
- **Définir les champs manuellement :** Dans ce cas, il faut définir le type de données manuellement (champs délimités ou de largeur fixe) et ensuite définir les noms de ces champs. Pour plus d'informations, voir la section Définition des champs.
- **Lire les champs dynamiquement :** Dans ce cas, le déclencheur peut recevoir des données structurées différemment. Par exemple, les nouveaux champs – La structure dynamique élimine le besoin de mettre à jour le filtre à chaque changement de structure. La prise en charge de la structure dynamique lit automatiquement tous les champs de données, même si de nouveaux champs existent ou si certains des anciens champs sont manquants. Elle les mappe automatiquement aux variables utilisant les mêmes noms. Pour plus d'informations, consulter l'article Activer la structure dynamique.

La section Aperçu de données simplifie la configuration. Dans le panneau de l'aperçu, le résultat d'un filtre défini est mis en évidence dans la zone d'aperçu à chaque changement de configuration. L'aperçu des données vous permet de vérifier quelles données sont extraites avec chaque règle.

2.1.2. Définition des champs

Pour les fichiers de texte structuré, la définition des champs est très simple. Il y a deux options :

- **Un séparateur définit les champs :** Dans ce cas, il y a un caractère de délimitation, tel que virgule ou point-virgule entre les champs. Il suffit de définir les noms de champs dans le même ordre d'apparition dans les données reçues par un déclencheur.
- **Champs de largeur fixe :** Dans ce cas, définir les noms de champs dans le même ordre d'apparition dans les données reçues par un déclencheur, et définir le nombre de caractères que le champ va occuper. Cette quantité de caractères sera lue dans les données destinées à ce champ.



Aperçu des données

Cette section fournit un aperçu de la définition du champ. Si l'élément défini est sélectionné, l'aperçu surligne son emplacement dans les données prévisualisées.

- **Aperçu du nom de fichier :** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si le nom du fichier d'aperçu est changé, le nouveau nom de fichier sera enregistré.
- **Ouvrir :** Sélectionne un autre fichier sur lequel les règles du filtre vont s'appliquer.
- **Actualiser :** Relance le filtre sur le contenu du fichier d'aperçu. Automation Met à jour la section Aperçu des données avec le résultat.

Options de formatage

Cette section définit les fonctions de manipulation de chaînes de caractères qui seront appliquées aux variables ou champs sélectionnés. Sélectionner une ou plusieurs fonctions. Ces fonctions s'appliquent dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début :** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin :** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Effacer le caractère d'ouverture et de fermeture :** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

Exemple

Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée **`{{selection}}`** est convertie en **`{selection}`**.

- **Rechercher et remplacer :** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *rechercher* et *remplacer par*. Vous pouvez aussi utiliser des expressions classiques.

> **NOTE**
> Il y a plusieurs implémentations des expressions classiques utilisées. utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consulter la Base de Connaissances article.

- **Remplacer les caractères non imprimables avec des espaces :** Remplace tous les caractères de contrôle de la chaîne par un espace (code décimal ASCII 32). Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Supprimer les caractères non imprimables :** Enlève tous les caractères de contrôle de la chaîne. Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux :** Décode les caractères (ou codes de contrôle) qui sont indisponibles sur le clavier, comme le retour chariot ou le passage à la ligne. utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Chariot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter la section Introduire des caractères spéciaux (Codes de Contrôle).
  Cette option convertit les caractères spéciaux de la syntaxe en caractères binaires réels.

Exemple

Quand vous recevez la séquence de données "<CR><LF>", les utilise comme une chaîne complète de 8 caractères. Activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires **CR** (Retour Chariot - code ASCII 13) et **LF** (Passage à la Ligne - code ASCII 10).

- **Rechercher et supprimer tout avant :** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et supprimer tout après :** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.
- **Changement de casse** : Modifie tous les caractères dans vos chaînes en lettres capitales ou minuscules.

2.1.3. Activer la structure dynamique

Le filtre de texte structuré peut identifier automatiquement les champs et leurs valeurs dans les données reçues. Ainsi, le mappage manuel de *variable au champ* n'est pas requis.

La fonction de **Structure dynamique** est utile si le déclencheur reçoit les données avec une structure changeante. Dans ce cas, la structure des données principale reste inchangée (par ex. : les champs sont délimités par une virgule) ou conserve la même structure, mais l'**ordre** et/ou le **nombre** de champs change. Il peut y avoir de nouveaux champs, ou certains des vieux champs ne sont plus disponibles. Avec la **Structure dynamique** activée, le filtre identifie automatiquement la structure du fichier reçu. Simultanément, le filtre lit les noms et les valeurs de champs (paires **`nom-valeur`**) à partir des données. Cela élimine le besoin de mappage manuel des champs vers les variables.

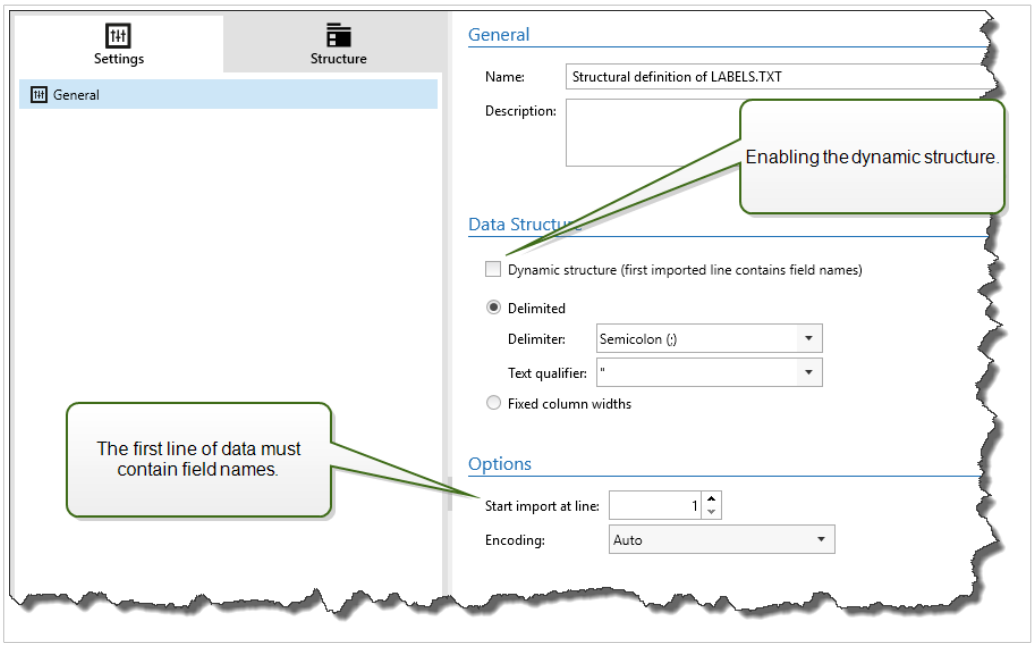
L'action Utiliser un filtre de données ne propose pas de possibilités de mappage, car elle effectue le mappage de manière dynamique. Il n'y a même pas besoin de définir les variables d'étiquettes dans la configuration du déclencheur. L'action assigne les valeurs de champs aux variables d'étiquettes de même nom sans avoir besoin des variables importées de l'étiquette. Toutefois, la règle s'applique seulement à l'action Imprimer l'étiquette. Pour utiliser les valeurs de champs dans une autre action, il faut définir les variables dans le déclencheur, tout en conservant le mappage automatique *variable-à-champ*.

NOTE

Aucune erreur n'est signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Ignore en silence les variables manquantes.

Configuration de la structure dynamique

Pour configurer la structure dynamique, activer l'option **Structure dynamique** dans les propriétés du Filtre de texte structuré.



- La première ligne de données doit contenir les noms de champs.
- La ligne sélectionnée pour **Commencer l'importation à la ligne** doit être la ligne contenant les noms de champs (généralement la première ligne dans les données).
- La structure de données doit être délimitée.
- Vous pouvez formater les données si nécessaire.

Options de formatage

Cette section définit les fonctions de manipulation de chaînes de caractères qui seront appliquées aux variables ou champs sélectionnés. Sélectionner une ou plusieurs fonctions. Ces fonctions s'appliquent dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début :** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin :** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Effacer le caractère d'ouverture et de fermeture :** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

Exemple

Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée `{{selection}}` est convertie en `{selection}`.

- **Rechercher et remplacer :** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *rechercher* et *remplacer par*. Vous pouvez aussi utiliser des expressions classiques.

> **NOTE**
> Il y a plusieurs implémentations des expressions classiques utilisées. utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consulter la Base de Connaissances article.

- **Remplacer les caractères non imprimables avec des espaces :** Remplace tous les caractères de contrôle de la chaîne par un espace (code décimal ASCII 32). Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Supprimer les caractères non imprimables :** Enlève tous les caractères de contrôle de la chaîne. Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux :** Décode les caractères (ou codes de contrôle) qui sont indisponibles sur le clavier, comme le retour chariot ou le passage à la ligne. utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Chariot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter la section Introduire des caractères spéciaux (Codes de Contrôle).
  Cette option convertit les caractères spéciaux de la syntaxe en caractères binaires réels.

Exemple

Quand vous recevez la séquence de données "<CR><LF>", les utilise comme une chaîne complète de 8 caractères. Activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires **CR** (Retour Chariot - code ASCII 13) et **LF** (Passage à la Ligne - code ASCII 10).

- **Rechercher et supprimer tout avant :** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et supprimer tout après :** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.
- **Changement de casse** : Modifie tous les caractères dans vos chaînes en lettres capitales ou minuscules.

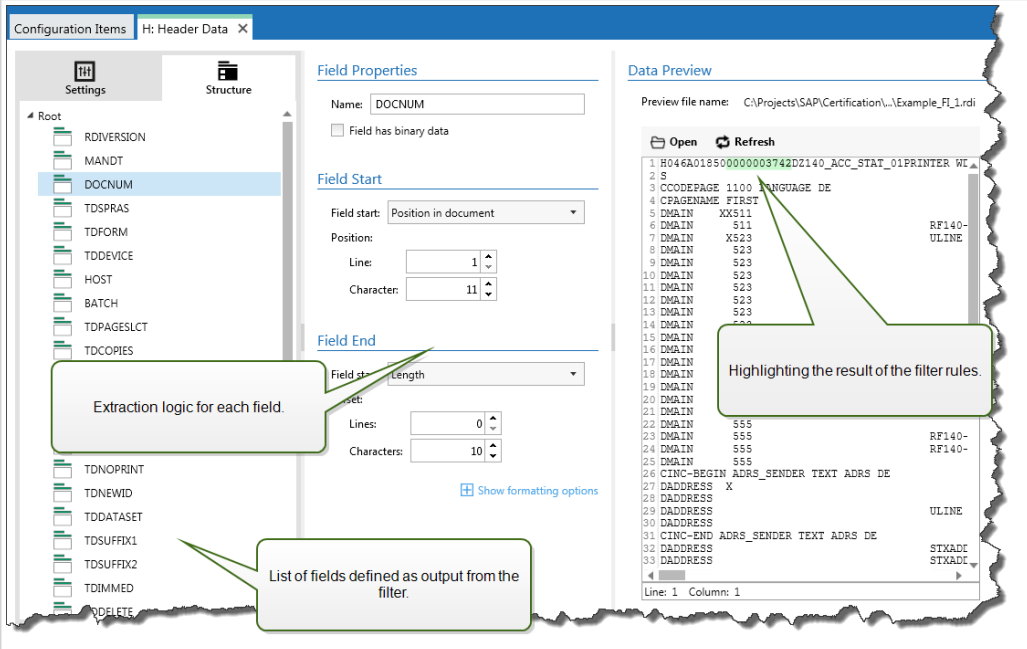
2.2. Filtre de données non structurées

2.2.1. Filtre de données non structurées

Pour en savoir plus sur les filtres en général, consulter l'article Comprendre les filtres.

Utilisez ce filtre chaque fois que le déclencheur reçoit des données non structurées, telles que des documents et rapports exportés d'anciens systèmes, de communication interceptée entre deux périphériques et d'une capture d'un flux d'impression. Le filtre vous permet d'extraire les champs individuels, les champs répétables dans les sous-zones, et même les paires **`nom-valeur`**.

Pour des exemples de données de texte structuré, voir les articles Données existantes, CSV composé et Fichiers binaires.



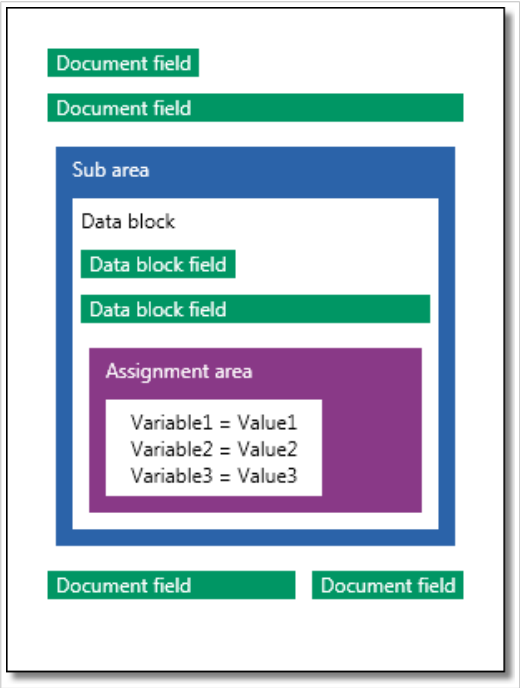
Définition de la structure

Éléments à utiliser pour configurer le filtre :

- **Champ :** Spécifie l'emplacement des données de champ entre le début et la fin du champ. Il y a différentes options pour définir l'emplacement du champ, du codage en dur au placement relatif. Il faut relier les champs définis aux variables respectives dans l'action Utiliser le Filtre de Données. Pour plus d'informations, voir la section Définition des champs.
- **Sous-zone :** Spécifie l'emplacement des données répétables. Chaque sous-zone définit au moins un bloc de données, qui à son tour contiendra des données pour les étiquettes. Il peut y avoir des sous-zones définies dans les sous-zones, permettant la définition de structures complexes. Des champs peuvent être définis dans chaque bloc de données. Il faut relier les champs définis aux variables respectives dans l'action. Pour chaque sous-zone, Automation définit un nouveau niveau d'espace réservé dans l'action Utiliser le Filtre de Données, permettant de mapper les variables aux champs de ce niveau. Pour plus d'informations, voir la section Définition des sous-zones.
- **Zone d'affectation :** Spécifie l'emplacement des données répétables contenant les paires **`nom-valeur`**. Automation lit les noms des champs et leurs valeurs simultanément. Automation effectue aussi automatiquement le mappage aux variables. Utiliser cette méthode pour adapter le filtre aux données d'entrée variables pour éliminer le temps de maintenance. Vous pouvez définir la zone d'affectation au niveau racine du document ou dans la sous-zone. Pour plus d'informations, voir la section Définition des zones d'affectation.

La section Aperçu de données simplifie la configuration. Le résultat des filtres définis est mis en évidence dans la zone d'aperçu à chaque changement de configuration. Vous pouvez visualiser les données extraites pour chaque règle.

Ce champ peut être défini au niveau racine comme champ du document. Les champs peuvent être définis dans un bloc de données. Les paires **`nom-valeur`** peuvent être définies dans la zone d'affectation.



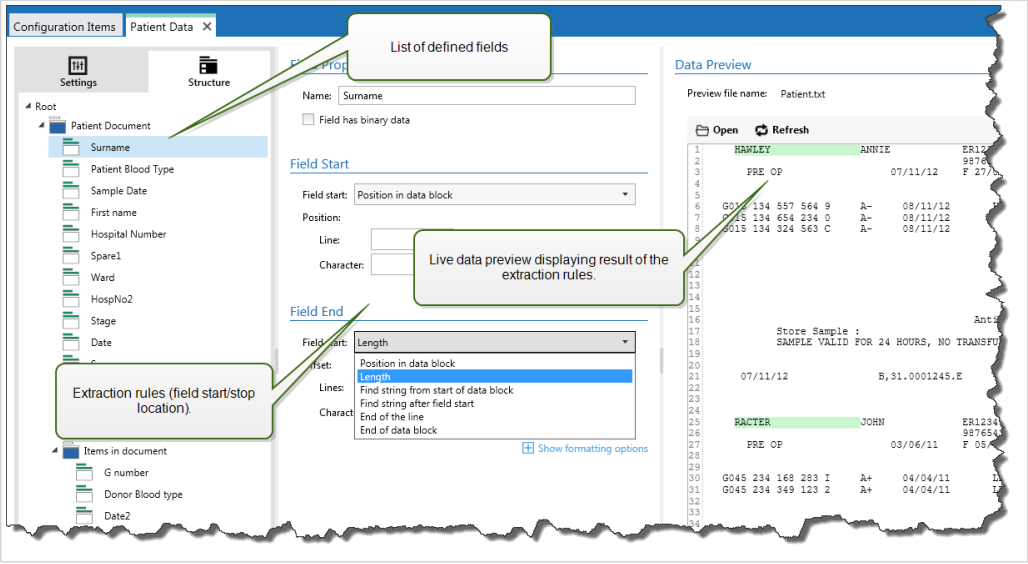
Général

Cette section définit les propriétés générales du filtre de données non structurées.

- **Nom :** Spécifie le nom du filtre. Utiliser un nom descriptif qui identifie le rôle d'un filtre dans une configuration. Il est modifiable à tout moment.
- **Description :** Vous permet de décrire l'objectif de ce filtre. Vous pouvez l'utiliser pour écrire une courte description de la fonction du filtre.
- **Encodage :** Spécifie l'encodage des données avec lesquelles ce filtre travaille.
- **Ignorer les lignes vides dans les blocs de données :** Spécifie de ne pas signaler d'erreur si le filtre extrait des valeurs de champs vides des blocs de données.

2.2.2. Définition des champs

Après avoir défini un champ, il faut définir son nom et ses règles d'extraction des valeurs du champ des données. Quand le filtre est exécuté, les règles d'extraction s'appliquent aux données d'entrée et les résultats assignés aux champs.



Propriétés des champs

- **Nom :** Spécifie le nom de champ unique.
- **Le champ contient des données binaires :** Spécifie que le champ a des données binaires. N'activer cette option que si les données attendues sont binaires.

Début de champ

- **Position dans le document :** La position codée en dur dans les données détermine le point de début/fin. L'origine des coordonnées est le coin en haut à gauche. Le caractère dans la position déterminée est inclus dans les données extraites.
- **Fin du document :** Le point de début/fin est à la fin du document. Vous pouvez aussi définir un décalage depuis la fin pour un nombre de lignes et/ou de caractères.
- **Rechercher une chaîne à partir du début du document :** La position de la chaîne de caractères recherchée définit le point de début/fin. Quand Automation a trouvé la chaîne de caractères requise, le caractère suivant détermine le point de début/fin. La chaîne de caractères recherchée n'est pas incluse dans les données extraites. La recherche par défaut est sensible à la casse.
    - **Commencer la recherche à partir de la position absolue :** Affiner la recherche en changeant la position de départ du début de donnée (position 1,1) avec un décalage. Utiliser cette fonctionnalité pour sauter la recherche au début des données.
    - **Occurrence :** Spécifie l'occurrence de la chaîne de caractères à retrouver. Utiliser cette option si après avoir trouvé la première chaîne, il ne faut pas attendre de déterminer la position début/fin.
    - **Décalage à partir de la chaîne :** Spécifie le décalage positif ou négatif après la chaîne de caractères recherchée.

Exemple

Vous pouvez définir le décalage pour inclure la chaîne recherchée dans les données extraites.

Fin de champ

- **Position dans le document :** La position codée en dur dans les données détermine le point de début/fin. L'origine des coordonnées est le coin en haut à gauche. Le caractère dans la position déterminée est inclus dans les données extraites.
- **Fin du document :** Le point de début/fin est à la fin du document. Vous pouvez aussi définir un décalage depuis la fin pour un nombre de lignes et/ou de caractères.
- **Rechercher une chaîne à partir du début du document :** La position de la chaîne de caractères recherchée définit le point de début/fin. Quand Automation a trouvé la chaîne de caractères requise, le caractère suivant détermine le point de début/fin. La chaîne de caractères recherchée n'est pas incluse dans les données extraites. La recherche par défaut est sensible à la casse.
    - **Commencer la recherche à partir de la position absolue :** Affiner la recherche en changeant la position de départ du début de donnée (position 1,1) avec un décalage. Utiliser cette fonctionnalité pour sauter la recherche au début des données.
    - **Occurrence :** Spécifie l'occurrence de la chaîne de caractères à retrouver. Utiliser cette option si après avoir trouvé la première chaîne, il ne faut pas attendre de déterminer la position début/fin.
    - **Décalage à partir de la chaîne :** Spécifie le décalage positif ou négatif après la chaîne de caractères recherchée.

    > **Exemple**
    >
    > Vous pouvez définir le décalage pour inclure la chaîne recherchée dans les données extraites.

- **Trouver la chaîne après le début du champ :** Le point de début/fin est défini par le point de départ de la recherche de chaîne de caractères comme dans l'option **Rechercher une chaîne à partir du début du document**, mais la recherche commence après la position du champ/zone, pas au début des données.
- **Longueur :** Spécifie la longueur des données en lignes ou en caractères. Le nombre de lignes et/ou de caractères spécifié sera extrait depuis la position de départ.
- **Fin de la ligne :** Spécifie d'extraire les données du point de départ jusqu'à la fin de la même ligne. Le décalage peut être négatif depuis la fin de la ligne.

Options de formatage

Cette section définit les fonctions de manipulation de chaînes de caractères qui seront appliquées aux variables ou champs sélectionnés. Sélectionner une ou plusieurs fonctions. Ces fonctions s'appliquent dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début :** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin :** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Effacer le caractère d'ouverture et de fermeture :** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

Exemple

Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée **`{{selection}}`** est convertie en **`{selection}`**.

- **Rechercher et remplacer :** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *rechercher* et *remplacer par*. Vous pouvez aussi utiliser des expressions classiques.

> **NOTE**
> Il y a plusieurs implémentations des expressions classiques utilisées. utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consulter la Base de Connaissances article.

- **Remplacer les caractères non imprimables avec des espaces :** Remplace tous les caractères de contrôle de la chaîne par un espace (code décimal ASCII 32). Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Supprimer les caractères non imprimables :** Enlève tous les caractères de contrôle de la chaîne. Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux :** Décode les caractères (ou codes de contrôle) qui sont indisponibles sur le clavier, comme le retour chariot ou le passage à la ligne. utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Chariot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter la section Introduire des caractères spéciaux (Codes de Contrôle).
  Cette option convertit les caractères spéciaux de la syntaxe en caractères binaires réels.

Exemple

Quand vous recevez la séquence de données "<CR><LF>", les utilise comme une chaîne complète de 8 caractères. Activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires **CR** (Retour Chariot - code ASCII 13) et **LF** (Passage à la Ligne - code ASCII 10).

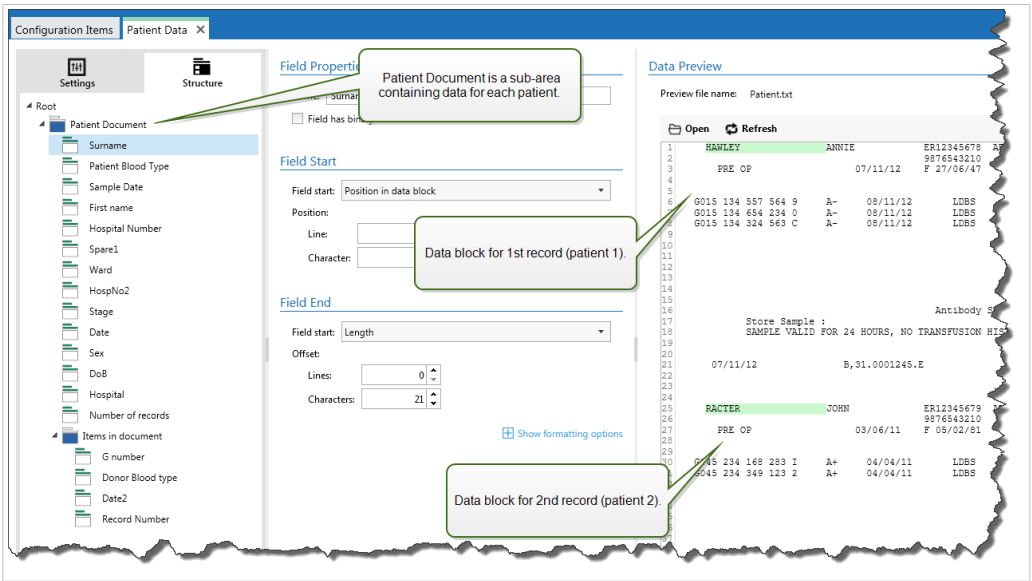
- **Rechercher et supprimer tout avant :** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et supprimer tout après :** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.
- **Changement de casse** : Modifie tous les caractères dans vos chaînes en lettres capitales ou minuscules.

2.2.3. Définition de sous-zones

Une sous-zone est la section de données dans laquelle plusieurs blocs de données sont identifiés par la même règle d'extraction. Chaque bloc de données fournit les données pour une seule étiquette.

Tous les blocs de données doivent être identifiés par la même règle de configuration. Chaque bloc de données peut contenir une autre sous-zone. Un nombre illimité de sous-zones indentées dans les sous-zones apparentées peut être déterminé.

Quand le filtre contient la définition d'une sous-zone, l'action Utiliser le Filtre de Données affiche les sous-zones avec les emplacements indentés. Toute action indentée sous une telle sous-zone s'exécute seulement pour les blocs de données de ce niveau. Différentes étiquettes vont s'imprimer avec les données de différentes sous-zones.



Configuration des sous-zones

La sous-zone est configurée comme les champs individuels. Chaque sous-zone est définie par les paramètres suivants.

- **Nom de la sous-zone :** Spécifie le nom de la sous-zone.
- **Blocs de données :** Spécifie la façon d'identifier les blocs de données dans la sous-zone. Chaque sous-zone contient au moins un bloc de données. Chaque bloc de données fournit les données pour une seule étiquette.
  - **Chaque bloc contient un nombre fixe de lignes :** Spécifie que chaque bloc de données dans une sous-zone contient un nombre fixe de lignes. Utiliser cette option quand chaque bloc de données contient exactement le même nombre de lignes.
  - **Les blocs commencent par une chaîne :** Spécifie que les blocs de données commencent par la chaîne de caractères fournie. Tout le contenu entre deux chaînes fournies appartient à un bloc de données distinct. Le contenu entre la dernière chaîne de caractères et la fin des données identifie le dernier bloc de données.
  - **Les blocs se terminent par une chaîne :** Spécifie que les blocs de données se terminent par la chaîne de caractères fournie. Tout le contenu entre deux chaînes fournies appartient à un bloc de données distinct. Le contenu entre le début des données et la première chaîne de caractères identifie le premier bloc de données.
  - **Les blocs sont séparés par une chaîne :** Spécifie que les blocs de données sont séparés par la chaîne de caractères sélectionnée. Tout le contenu entre les deux chaînes de caractères sélectionnées appartient à un bloc de données distinct.
- **Début du premier bloc de données :** Spécifie le point de départ du premier bloc de données. Simultanément, définit le point de départ de la sous-zone. Généralement, le point de départ est le début des données reçues. Les paramètres de configuration sont les mêmes que pour la définition des champs. Pour plus d'informations, voir la section Définition des champs.
- **Fin du dernier bloc de données :** Spécifie la position de fin du dernier bloc de données. Simultanément, définit la position de fin de la sous-zone. Généralement, la position de fin est la fin des données reçues. Les paramètres de configuration sont les mêmes que pour la définition des champs. Pour plus d'informations, voir la section Définition des champs.

Configuration des champs dans la sous-zone

Dans la sous-zone, les champs sont configurés en utilisant les mêmes paramètres que pour les champs définis au niveau de la racine. Pour plus d'informations, voir la section Définition des champs.

NOTE

Les numéros des lignes de champs se réfèrent à la position dans le bloc de données, pas à la position dans les données d'entrée.

Aperçu des données

Cette section fournit un aperçu de la définition du champ. Si l'élément défini est sélectionné, l'aperçu surligne son emplacement dans les données prévisualisées.

- **Aperçu du nom de fichier :** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si le nom du fichier d'aperçu est changé, le nouveau nom de fichier sera enregistré.
- **Ouvrir :** Sélectionne un autre fichier sur lequel les règles du filtre vont s'appliquer.
- **Actualiser :** Relance le filtre sur le contenu du fichier d'aperçu. Automation Met à jour la section Aperçu des données avec le résultat.

2.2.4. Définition des zones d'affectation

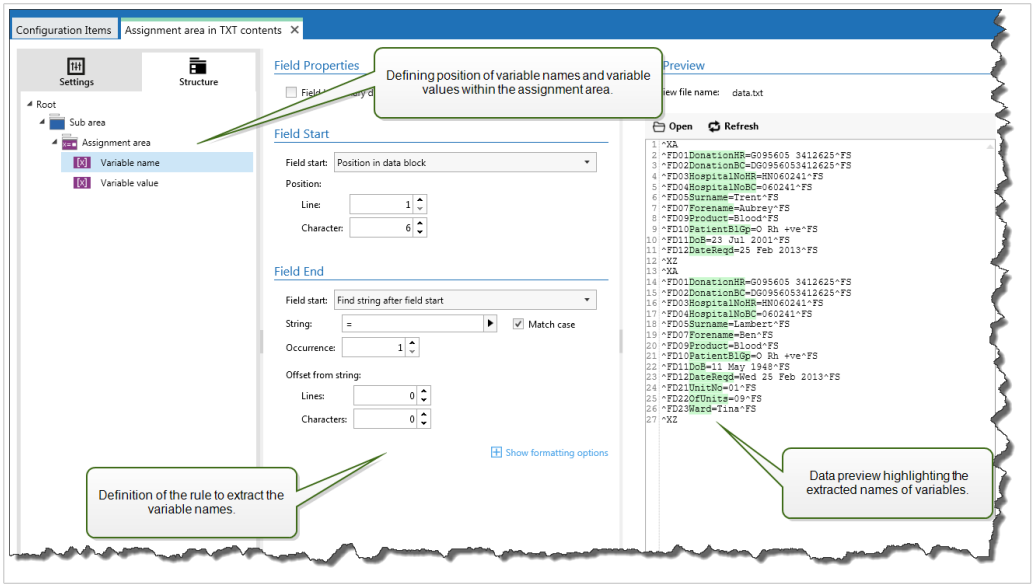
Le filtre de données non structurées identifie automatiquement les champs et leurs valeurs dans les données reçues. Ainsi, le mappage manuel de *variable au champ* n'est pas requis.

La fonction de **Structure dynamique** est utile si le déclencheur reçoit les données avec une structure changeante. Dans ce cas, la structure des données principale reste inchangée (par ex. : les champs sont délimités par une virgule) ou conserve la même structure, mais l'**ordre** et/ou le **nombre** de champs change. Il peut y avoir de nouveaux champs, ou certains des vieux champs ne sont plus disponibles. Avec la **Structure dynamique** activée, le filtre identifie automatiquement la structure du fichier reçu. Simultanément, le filtre lit les noms et les valeurs de champs (paires **`nom-valeur`**) à partir des données. Cela élimine le besoin de mappage manuel des champs vers les variables.

L'action Utiliser un filtre de données ne propose pas de possibilités de mappage, car elle effectue le mappage de manière dynamique. Il n'y a même pas besoin de définir les variables d'étiquettes dans la configuration du déclencheur. L'action assigne les valeurs de champs aux variables d'étiquettes de même nom sans avoir besoin des variables importées de l'étiquette. Toutefois, la règle s'applique seulement à l'action Imprimer l'étiquette. Pour utiliser les valeurs de champs dans une autre action, il faut définir les variables dans le déclencheur, tout en conservant le mappage automatique *variable-à-champ*.

NOTE

Aucune erreur n'est signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Ignore en silence les variables manquantes.



Configuration de la zone d'affectation

La zone d'affectation est configurée en utilisant la même procédure que pour la sous-zone. Pour plus d'informations, voir la section Définition des sous-zones. La zone d'affectation peut être définie au niveau des données racine, apparaissant une seule fois. Ou elle peut être configurée dans une sous-zone. Elle sera ainsi exécutée pour chaque bloc de données dans la sous-zone.

Configuration des champs dans la zone d'affectation

Lorsque vous créez une zone d'affectation, le filtre définit automatiquement deux espaces réservés. Ces deux espaces réservés définissent la paire **`nom-valeur`**.

- **Nom de variable :** Spécifie le champ dont le contenu sera le nom de variable (composant **`nom`** en paire). Configurer le champ en utilisant la même procédure que pour les champs de document. Pour plus d'informations, voir la section Définition des champs.
- **Valeur de variable :** Spécifie le champ dont le contenu sera la valeur de variable (composant **`valeur`** en paire). Configurer le champ en utilisant la même procédure que pour les champs de document. Pour plus d'informations, voir la section Définition des champs.

Exemple

La zone entre ^XA et ^XZ est la zone d'affectation. Chaque ligne de la zone d'affectation fournit les paires **`nom-valeur`**. Le **Nom** est défini comme une valeur entre le 6e caractère de la ligne et le signe égal. La **Valeur** est définie comme la valeur entre le signe égal et la fin de la ligne, avec un décalage négatif de trois caractères.

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

2.3. Configuration du filtre XML

2.3.1. Filtre XML

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Pour en savoir plus sur les filtres en général, consulter l'article Comprendre les filtres.

Utiliser ce filtre quand un déclencheur reçoit des données encodées XML. Le filtre vous permet d'extraire les champs individuels, les champs répétables dans les sous-zones, et même les paires **`nom-valeur`**. La structure XML définit les éléments et sous-éléments, attributs et leurs valeurs et les valeurs texte (valeurs d'élément).

Vous pouvez définir la structure d'un fichier XML par vous-même, mais NiceLabel vous recommande de l'importer du modèle de fichier XML donné. Cliquer sur le bouton **Importer la structure de données** dans le ruban. Après avoir importé la structure XML, la section Aperçu des données affiche le contenu XML. Elle met aussi en évidence les éléments et attributs que vous définissez en tant que champs de sortie.

Pour les exemples de données XML, voir l'article Données XML.

Définition de la structure

Pour utiliser les éléments XML, les configurer comme suit :

- **Valeur de variable :** Spécifie que les éléments sélectionnés sont des champs dont les valeurs vont être associées aux variables correspondantes par l'action Utiliser le Filtre de Données. Pour plus d'informations, voir la section Définition des champs XML.
    - **Élément optionnel :** Spécifie que cet élément n'est pas obligatoire. Cela correspond à l'attribut du schéma XML (fichier XSD) `minOccurs=0`. La variable mappée à ce type de champ aura une valeur vide si l'élément n'apparaît pas dans l'XML.
- **Bloc de données :** Spécifie que l'élément sélectionné survient plusieurs fois et fournit des données pour une seule étiquette. Le bloc de données peut être défini comme zone répétable, comme zone d'affectation ou les deux.
    - **Zone répétable :** Spécifie que les valeurs de tous les blocs de données répétables seront extraites, pas seulement celles du premier bloc. Des champs peuvent être définis dans chaque bloc de données. Il faut mapper les champs définis aux variables respectives dans l'action Utiliser le Filtre de Données. Pour plus d'informations, voir la section Définir les éléments répétables.
    - **Zone d'affectation :** Spécifie que le bloc de données contient des paires `nom-valeur`. Les noms de champs et leurs valeurs sont lus simultanément. Le mappage aux variables est effectué automatiquement. Utiliser cette méthode pour adapter le filtre aux données d'entrée variables pour éliminer le temps de maintenance. Pour plus d'informations, voir la section Définition de la zone d'affectation XML.

La section Aperçu de données simplifie la configuration. Le résultat d'un filtre défini est mis en évidence dans la zone d'aperçu.

Pour changer les données XML de l'aperçu, cliquer sur **Ouvrir** et rechercher un nouvel exemple de fichier XML.

2.3.2. Définition des champs XML

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Quand un champ XML est défini, les valeurs des éléments sélectionnés deviennent automatiquement disponibles. La définition d'un filtre rend ces champs disponibles pour le mappage des variables aux actions. Vous pouvez ainsi extraire les valeurs des éléments ou attributs.

Pour définir la valeur de l'élément comme champ, procéder comme suit :

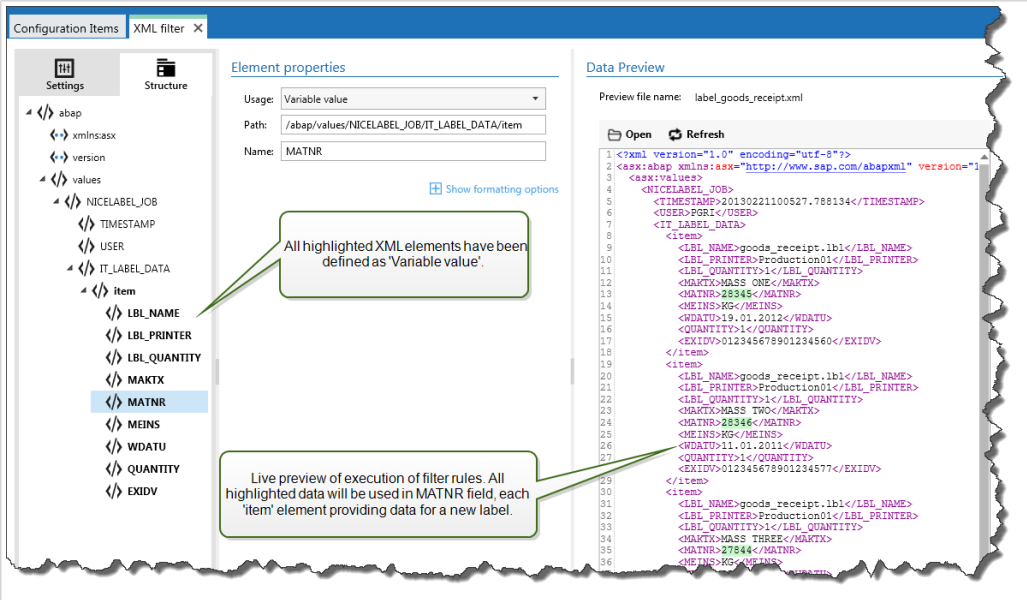
1. Sélectionner l'élément ou attribut dans la liste de structure.

2. Pour **Usage**, sélectionner **Valeur de variable**.

3. L'élément s'affiche en gras sur la liste de structure, ce qui indique qu'il est en cours d'utilisation.

4. L'élément ou nom d'attribut est utilisé comme nom du champ de sortie.

5. La section Aperçu de données met en évidence la valeur de l'élément sélectionné.



Options de formatage

Cette section définit les fonctions de manipulation de chaînes de caractères qui seront appliquées aux variables ou champs sélectionnés. Sélectionner une ou plusieurs fonctions. Ces fonctions s'appliquent dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début :** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin :** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Effacer le caractère d'ouverture et de fermeture :** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

Exemple

Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée **`{{selection}}`** est convertie en **`{selection}`**.

- **Rechercher et remplacer :** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *rechercher* et *remplacer par*. Vous pouvez aussi utiliser des expressions classiques.

> **NOTE**
> Il y a plusieurs implémentations des expressions classiques utilisées. utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consulter la Base de Connaissances article.

- **Remplacer les caractères non imprimables avec des espaces :** Remplace tous les caractères de contrôle de la chaîne par un espace (code décimal ASCII 32). Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Supprimer les caractères non imprimables :** Enlève tous les caractères de contrôle de la chaîne. Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux :** Décode les caractères (ou codes de contrôle) qui sont indisponibles sur le clavier, comme le retour chariot ou le passage à la ligne. utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Chariot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter la section Introduire des caractères spéciaux (Codes de Contrôle).
  Cette option convertit les caractères spéciaux de la syntaxe en caractères binaires réels.

Exemple

Quand vous recevez la séquence de données "<CR><LF>", les utilise comme une chaîne complète de 8 caractères. Activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires **CR** (Retour Chariot - code ASCII 13) et **LF** (Passage à la Ligne - code ASCII 10).

- **Rechercher et supprimer tout avant :** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et supprimer tout après :** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.
- **Changement de casse** : Modifie tous les caractères dans vos chaînes en lettres capitales ou minuscules.

Aperçu des données

Cette section fournit un aperçu de la définition du champ. Si l'élément défini est sélectionné, l'aperçu surligne son emplacement dans les données prévisualisées.

- **Aperçu du nom de fichier :** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si le nom du fichier d'aperçu est changé, le nouveau nom de fichier sera enregistré.
- **Ouvrir :** Sélectionne un autre fichier sur lequel les règles du filtre vont s'appliquer.
- **Actualiser :** Relance le filtre sur le contenu du fichier d'aperçu. Automation Met à jour la section Aperçu des données avec le résultat.

2.3.3. Définir les éléments répétables dans le Filtre XML

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Si un élément XML survient dans les données XML plusieurs fois, c'est un élément répétable. Généralement, un élément répétable contient des données pour une seule étiquette. Pour utiliser les données de tous les éléments répétables, pas seulement le premier, il faut définir l'élément comme un **bloc de données** et activer l'option **Élément répétable**. Si le filtre contient une définition d'éléments définis comme bloc de données/élément répétable, l'action Utiliser le Filtre de Données affiche les éléments répétables avec les espaces réservés indentés. Toute action indentée sous un tel espace réservé s'exécute seulement pour les blocs de données à ce niveau.

Exemple

L'élément **<item>** est défini comme un **bloc de Données** et **Élément répétable**. Il indique au filtre d'extraire toutes les occurrences de l'élément **<item>**, pas seulement le premier. Dans ce cas, l'élément **<item>** doit être défini comme le sous-niveau dans l'action **Utiliser le Filtre de Données**. Vous devez indenter les actions Ouvrir l'étiquette et Imprimer l'étiquette sous cet espace réservé de sous-niveau, pour qu'elles soient bouclées autant de fois qu'il y a d'occurrences de l'élément **<item>**. Comme montré dans l'exemple ci-dessous, à trois reprises.

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

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

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

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

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

AVERTISSEMENT

Le filtre de données xml avec un bloc de données imbriquées duplique les valeurs provenant de la zone d'affectation précédente si une variable est définie dans le premier bloc, mais pas dans le deuxième. Prenons par exemple une situation où vous imprimez deux étiquettes. Dans la première zone d'affectation, vous avez défini une variable **`<manufacturer_serial>`**. Dans la deuxième zone d'affectation, cette variable n'est pas présente.

```
<?xml ver sion="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" ver sion="1.0">
    <jobs>
      <element>
         <data>
            <item>
               <manufacturer_serial>706583012001</
manufacturer_serial>
               <package_code>AA792CY</package_code>
            </item>
         </data>
         <label_filename>serial_number_label.nlbl</label_filename>
         <print_quantity>1</print_quantity>
         <printer_name>Warehouse_printer_09</printer_name>
      </element>
      <element>
         <data>
            <item>
               <package_code>BE273XS</package_code>
            </item>
         </data>
         <label_filename>serial_number_label.nlbl</label_filename>
         <print_quantity>1</print_quantity>
         <printer_name>Warehouse_09</printer_name>
      </element>
   </jobs>
</asx:abap>
```

Le résultat attendu serait la deuxième étiquette sans les données **`<manufacturer_serial>`**. Mais sur l'impression obtenue, la deuxième étiquette contiendrait la valeur **`<manufacturer_serial>`** provenant de la première étiquette, à savoir 706583012001.

Quatre solutions sont possibles :

- Éviter des blocs de données imbriquées.
- Définir des valeurs de variables vides si les variables ne sont pas utilisées dans les blocs de données imbriquées.

```
<item>
   <manufacturer_serial></manufacturer_serial>
   <package_code>AA792CY</package_code>
</item>
```

- Utiliser des conditions sur les actions pour définir les variables.
- Utiliser des manipulations de chaîne lorsque des variables manquantes ont été détectées.

2.3.4. Définition de la zone d'affectation XML

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

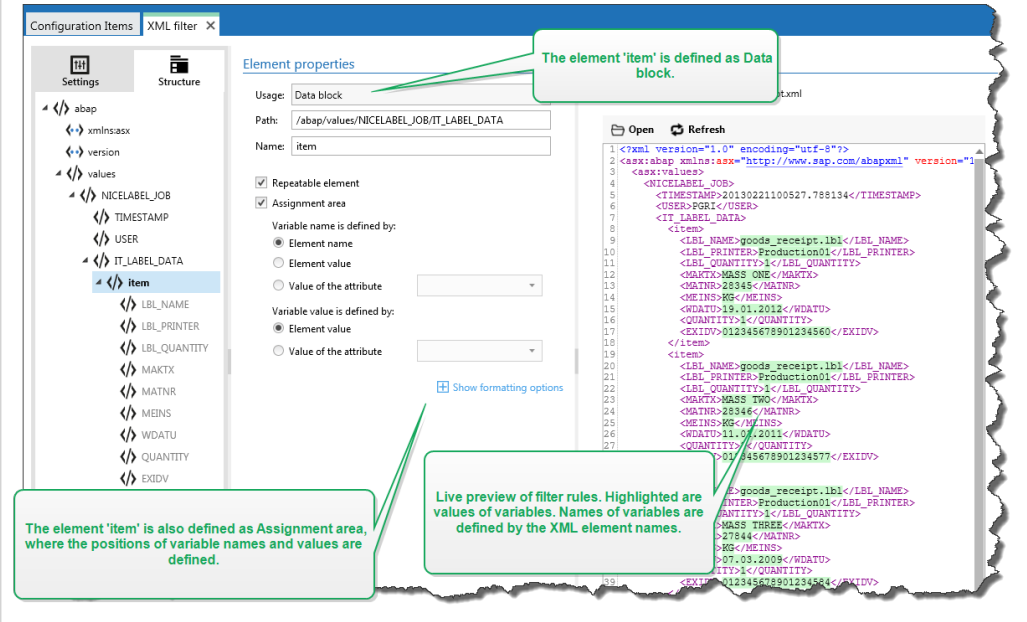
Le filtre XML identifie automatiquement les champs et leurs valeurs dans les données reçues. Ainsi, le mappage manuel de *variable au champ* n'est pas requis.

La fonction de **Structure dynamique** est utile si le déclencheur reçoit les données avec une structure changeante. Dans ce cas, la structure des données principale reste inchangée (par ex. : les champs sont délimités par une virgule) ou conserve la même structure, mais l'**ordre** et/ou le **nombre** de champs change. Il peut y avoir de nouveaux champs, ou certains des vieux champs ne sont plus disponibles. Avec la **Structure dynamique** activée, le filtre identifie automatiquement la structure du fichier reçu. Simultanément, le filtre lit les noms et les valeurs de champs (paires **`nom-valeur`**) à partir des données. Cela élimine le besoin de mappage manuel des champs vers les variables.

L'action Utiliser un filtre de données ne propose pas de possibilités de mappage, car elle effectue le mappage de manière dynamique. Il n'y a même pas besoin de définir les variables d'étiquettes dans la configuration du déclencheur. L'action assigne les valeurs de champs aux variables d'étiquettes de même nom sans avoir besoin des variables importées de l'étiquette. Toutefois, la règle s'applique seulement à l'action Imprimer l'étiquette. Pour utiliser les valeurs de champs dans une autre action, il faut définir les variables dans le déclencheur, tout en conservant le mappage automatique *variable-à-champ*.

NOTE

Aucune erreur n'est signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Ignore en silence les variables manquantes.



Configuration de la zone d'affectation XML

Quand un bloc de données est défini comme zone d'affectation, deux espaces réservés apparaissent sous la définition de cet élément. Vous devez définir comment le nom de champ et sa valeur sont définis pour permettre au filtre d'extraire les paires **`nom-valeur`**.

- **Nom de variable :** Spécifie l'élément qui contient le nom de champ. Le nom peut être défini par le nom d'élément, la valeur de l'attribut sélectionné ou la valeur de l'élément. Pour activer le mappage automatique, la variable de l'étiquette doit avoir le même nom.
- **Valeur de variable :** Spécifie l'élément qui contient la valeur du champ. Le nom peut être défini par le nom d'élément, la valeur de l'attribut sélectionné ou la valeur de l'élément.

AVERTISSEMENT

L'élément XML qui contient les paires **`nom-valeur`** ne peut pas être l'élément racine, mais doit au moins être un élément de second niveau. Par exemple, dans l'élément XML ci-dessous, l'élément **`<label>`** est l'élément de second niveau et peut contenir les paires **`nom-valeur`**.

Options de formatage

Cette section définit les fonctions de manipulation de chaînes de caractères qui seront appliquées aux variables ou champs sélectionnés. Sélectionner une ou plusieurs fonctions. Ces fonctions s'appliquent dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début :** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin :** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Effacer le caractère d'ouverture et de fermeture :** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

Exemple

Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée **`{{selection}}`** est convertie en **`{selection}`**.

- **Rechercher et remplacer :** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *rechercher* et *remplacer par*. Vous pouvez aussi utiliser des expressions classiques.

  NOTE

  Il y a plusieurs implémentations des expressions classiques utilisées. utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consulter la Base de Connaissances article.

- **Remplacer les caractères non imprimables avec des espaces :** Remplace tous les caractères de contrôle de la chaîne par un espace (code décimal ASCII 32). Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Supprimer les caractères non imprimables :** Enlève tous les caractères de contrôle de la chaîne. Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux :** Décode les caractères (ou codes de contrôle) qui sont indisponibles sur le clavier, comme le retour chariot ou le passage à la ligne. utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Chariot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter la section Introduire des caractères spéciaux (Codes de Contrôle).
  Cette option convertit les caractères spéciaux de la syntaxe en caractères binaires réels.

Exemple

Quand vous recevez la séquence de données "<CR><LF>", les utilise comme une chaîne complète de 8 caractères. Activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires **CR** (Retour Chariot - code ASCII 13) et **LF** (Passage à la Ligne - code ASCII 10).

- **Rechercher et supprimer tout avant :** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et supprimer tout après :** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.
- **Changement de casse** : Modifie tous les caractères dans vos chaînes en lettres capitales ou minuscules.

Exemple

L'élément **`<label>`** est défini comme bloc de données et zone d'affectation. Le **nom de variable** est défini par la valeur du nom d'attribut, la **valeur de variable** est définie par l'élément texte.

```
<?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>
```

Vérifier également l'échantillon NiceLabel Automation dans le dossier **`C:\Users\Public\Documents\NiceLabel 10\Automation\Samples\XML Assignment Area`**.

2.4. Configuration du filtre JSON

2.4.1. Filtre JSON

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Pour en savoir plus sur les filtres en général, consulter l'article Comprendre les filtres.

Utiliser le filtre JSON quand un déclencheur reçoit des données encodées JSON. Le filtre JSON vous permet d'utiliser les variables et les valeurs de votre fichier JSON. Le filtre prend en charge l'extraction des données des spectres JSON.

> **NOTE**
> Automation vous permet d'utiliser tous les types de données JSON. Plus d'informations sur les types de données JSON ici.

Vous pouvez définir la structure d'un fichier JSON manuellement, mais NiceLabel vous recommande de l'importer des fichiers JSON reçus.

Pour importer la structure d'un fichier JSON :

1. Aller dans **Filtre de données** et **Modifier** votre filtre JSON.
2. Cliquer sur **Structure** > **Importer la structure des données**. Parcourir votre fichier JSON et cliquer sur **Ouvrir**.
   Après avoir importé les fichiers JSON, la section **Aperçu de données** affiche le contenu JSON. L'Aperçu des données surligne également les éléments que vous définissez comme champs de sortie.

Pour les exemples de données JSON, voir la section Données JSON.

Définition de la structure

Pour utiliser les éléments JSON, les configurer comme suit :

- **Valeur de variable :** Spécifie que vous voulez utiliser l'élément sélectionné en tant que champ. En concevant la configuration, vous mappez manuellement avec les variables respectives dans l'action Utiliser un Filtre de Données. Pour plus d'informations, voir la section Définition des champs JSON.
    - **Élément optionnel** : Spécifie que cet élément n'est pas obligatoire. La variable mappée à ce type de champ aura une valeur vide si l'élément n'apparaît pas dans le fichier JSON.
- **Bloc de données :** Spécifie que les sous-éléments inclus surviennent plusieurs fois et fournissent des données pour vos étiquettes. Le bloc de données peut être défini comme zone répétable, comme zone d'affectation ou les deux. En termes de JSON, le Bloc de données fonctionne comme un spectre.
    - **Zone répétable :** Spécifie que les valeurs de tous les blocs de données répétables seront extraites, pas seulement celles du premier bloc. Des champs peuvent être définis dans chaque bloc de données. Mapper manuellement les champs définis aux variables respectives dans l'action Utiliser le Filtre de Données. Pour plus d'informations, voir la section Définir les éléments répétables dans le Filtre JSON.
    - **Zone d'affectation :** Crée automatiquement les variables et leur assigne les valeurs pertinentes. Les noms de champs et leurs valeurs sont lus simultanément. Le mappage aux variables est effectué automatiquement. Utiliser cette méthode pour adapter le filtre aux données d'entrée variables pour éliminer le temps de maintenance. Pour plus d'informations, voir la section Définition de la zone d'affectation JSON.

La section **Aperçu de données** simplifie la configuration. Le résultat d'un filtre défini est mis en évidence dans la zone d'aperçu.

Pour changer les données JSON de l'aperçu, cliquer sur **Ouvrir** et rechercher un nouvel exemple de fichier JSON.

2.4.2. Définition des champs JSON

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Quand un champ JSON est défini, les valeurs des éléments sélectionnés deviennent automatiquement disponibles. Votre définition du filtre rend ces champs disponibles pour le mappage des variables aux actions. Vous pouvez ainsi extraire les valeurs des éléments.

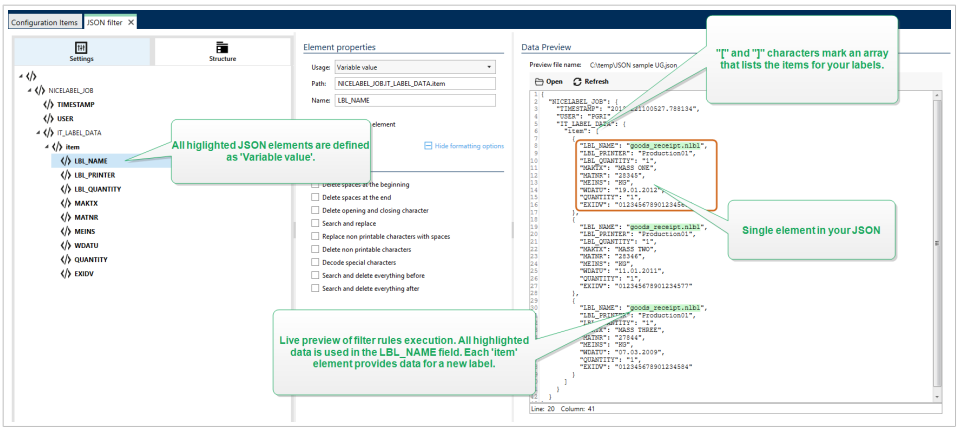
Pour définir les champs JSON :

1. Sélectionner votre élément et définir son **Usage** sur **Valeur variable**.

2. L'élément s'affiche en gras sur la liste de structure, ce qui indique qu'il est en cours d'utilisation.

3. L'élément est utilisé comme nom du champ de sortie.

4. La section Aperçu de données met en évidence les valeurs de l'élément sélectionné.



Options de formatage

Cette section définit les fonctions de manipulation de chaînes de caractères qui seront appliquées aux variables ou champs sélectionnés. Sélectionner une ou plusieurs fonctions. Ces fonctions s'appliquent dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début :** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin :** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Effacer le caractère d'ouverture et de fermeture :** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

Exemple

Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée **`{{selection}}`** est convertie en **`{selection}`**.

- **Rechercher et remplacer :** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *rechercher* et *remplacer par*. Vous pouvez aussi utiliser des expressions classiques.

> **NOTE**
> Il y a plusieurs implémentations des expressions classiques utilisées. utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consulter la Base de Connaissances article.

- **Remplacer les caractères non imprimables avec des espaces :** Remplace tous les caractères de contrôle de la chaîne par un espace (code décimal ASCII 32). Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Supprimer les caractères non imprimables :** Enlève tous les caractères de contrôle de la chaîne. Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux :** Décode les caractères (ou codes de contrôle) qui sont indisponibles sur le clavier, comme le retour chariot ou le passage à la ligne. utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Chariot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter la section Introduire des caractères spéciaux (Codes de Contrôle).
  Cette option convertit les caractères spéciaux de la syntaxe en caractères binaires réels.

Exemple

Quand vous recevez la séquence de données "<CR><LF>", les utilise comme une chaîne complète de 8 caractères. Activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires **CR** (Retour Chariot - code ASCII 13) et **LF** (Passage à la Ligne - code ASCII 10).

- **Rechercher et supprimer tout avant :** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et supprimer tout après :** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.
- **Changement de casse** : Modifie tous les caractères dans vos chaînes en lettres capitales ou minuscules.

Aperçu des données

Cette section fournit un aperçu de la définition du champ. Si l'élément défini est sélectionné, l'aperçu surligne son emplacement dans les données prévisualisées.

- **Aperçu du nom de fichier :** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si le nom du fichier d'aperçu est changé, le nouveau nom de fichier sera enregistré.
- **Ouvrir :** Sélectionne un autre fichier sur lequel les règles du filtre vont s'appliquer.
- **Actualiser :** Relance le filtre sur le contenu du fichier d'aperçu. Automation Met à jour la section Aperçu des données avec le résultat.

2.4.3. Définir les éléments répétables dans le Filtre JSON

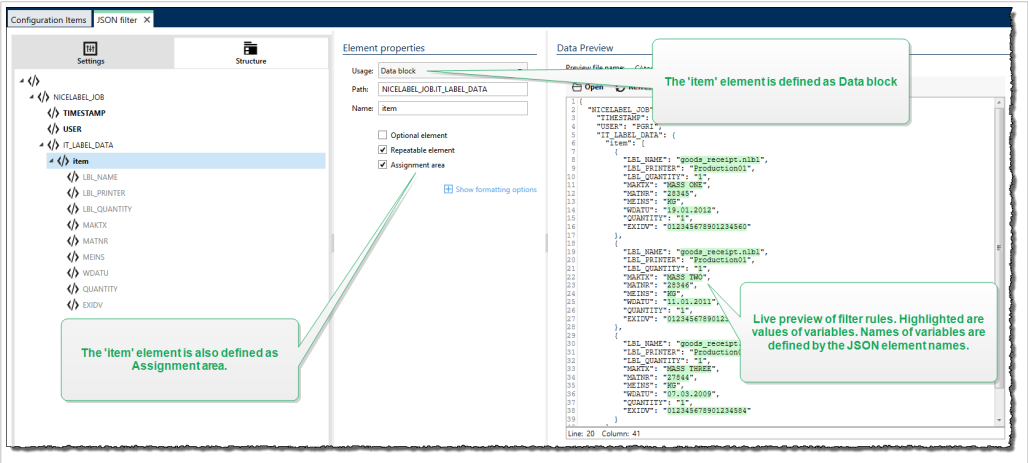
INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Si un élément JSON survient dans les données JSON plusieurs fois, c'est un élément répétable. Généralement, un élément répétable contient des données pour un seul masque d'étiquette. Les éléments répétables produisent plusieurs étiquettes remplies des données pertinentes.

Pour indiquer que vous voulez utiliser les valeurs de tous les éléments répétables, pas seulement ceux du premier :

1. Sélectionner l'élément et le définir comme **Bloc de données**.
2. Activer l'option **Élément répétable**.



Si le filtre contient une définition d'éléments définis comme bloc de données/élément répétable, l'action Utiliser le Filtre de Données affiche les éléments répétables avec les espaces réservés indentés. Toute action indentée sous un tel espace réservé s'exécute seulement pour les blocs de données à ce niveau.

Exemple

L'élément **`item`** est défini comme un **bloc de Données** et comme un **Élément répétable**. Il indique au filtre d'extraire toutes les occurrences du spectre, pas seulement le premier. Dans ce cas, l'élément **item** doit être défini comme le sous-niveau dans l'action **Utiliser le Filtre de Données**. Vous devez indenter les actions Ouvrir l'étiquette et Imprimer l'étiquette sous cet espace réservé de sous-niveau, pour qu'elles soient bouclées autant de fois qu'il y a d'occurrences de l'élément **`item`**. Comme montré dans l'exemple ci-dessous, à trois reprises.

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

```
                }
            ]
        }
    }
}
```

AVERTISSEMENT

Le filtre de données json avec un bloc de données imbriquées duplique les valeurs provenant de la zone d'affectation précédente si une variable est définie dans le premier bloc, mais pas dans le deuxième. Prenons par exemple une situation où vous imprimez deux étiquettes. Dans la première zone d'affectation, vous avez défini une variable **« `manufacturer_serial` »**. Dans la deuxième zone d'affectation, cette variable n'est pas présente.

```
{
  "jobs": [
    {
      "data": [
        {
          "manufacturer_serial": "706583012001",
          "package_code": "AA792CY"
        }
      ],
      "label_filename": "serial_number_label.nlbl",
      "print_quantity": 1,
      "printer_name": "Warehouse_printer_09"
    },
    {
      "data": [
        {
          "package_code": "BE273XS"
        }
      ],
      "label_filename": "serial_number_label.nlbl",
      "print_quantity": 1,
      "printer_name": "Warehouse_09"
    }
  ]
}
```

Le résultat attendu serait la deuxième étiquette sans les données **« `manufacturer_serial` »**. Mais sur l'impression obtenue, la deuxième étiquette contiendrait la valeur **« `manufacturer_serial` »** provenant de la première étiquette, à savoir 706583012001.

Trois solutions sont possibles :

- Éviter des blocs de données imbriquées.
- Définir des valeurs de variables vides (« ») si les variables ne sont pas utilisées dans les blocs de données imbriquées.

```
      "data": [
        {
```

```
        "manufacturer_serial": "",
        "package_code": "AA792CY"
    }
```

- Utiliser des conditions sur les actions pour définir les variables.
- Utiliser des manipulations de chaîne lorsque des variables manquantes ont été détectées.

2.4.4. Définition de la zone d'affectation JSON

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

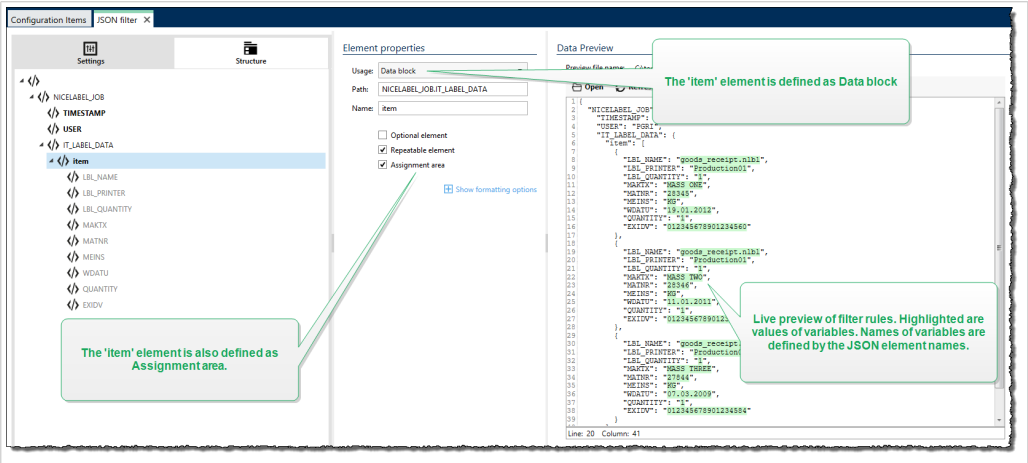
Le filtre JSON identifie automatiquement les champs et leurs valeurs dans les données reçues. Ainsi, le mappage manuel de *variable au champ* n'est pas requis.

La fonction de **Structure dynamique** est utile si le déclencheur reçoit les données avec une structure changeante. Dans ce cas, la structure des données principale reste inchangée (par ex. : les champs sont délimités par une virgule) ou conserve la même structure, mais l'**ordre** et/ou le **nombre** de champs change. Il peut y avoir de nouveaux champs, ou certains des vieux champs ne sont plus disponibles. Avec la **Structure dynamique** activée, le filtre identifie automatiquement la structure du fichier reçu. Simultanément, le filtre lit les noms et les valeurs de champs (paires **`nom-valeur`**) à partir des données. Cela élimine le besoin de mappage manuel des champs vers les variables.

L'action Utiliser un filtre de données ne propose pas de possibilités de mappage, car elle effectue le mappage de manière dynamique. Il n'y a même pas besoin de définir les variables d'étiquettes dans la configuration du déclencheur. L'action assigne les valeurs de champs aux variables d'étiquettes de même nom sans avoir besoin des variables importées de l'étiquette. Toutefois, la règle s'applique seulement à l'action Imprimer l'étiquette. Pour utiliser les valeurs de champs dans une autre action, il faut définir les variables dans le déclencheur, tout en conservant le mappage automatique *variable-à-champ*.

NOTE

Aucune erreur n'est signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Ignore en silence les variables manquantes.



NOTE

Étant donné qu'il n'y a pas d'attributs optionnels dans JSON, Automation définit automatiquement les **noms des variables** et les **valeurs des variables**.

Options de formatage

Cette section définit les fonctions de manipulation de chaînes de caractères qui seront appliquées aux variables ou champs sélectionnés. Sélectionner une ou plusieurs fonctions. Ces fonctions s'appliquent dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début :** Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin :** Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Effacer le caractère d'ouverture et de fermeture :** Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

Exemple

Si vous utilisez "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée **`{{selection}}`** est convertie en **`{selection}`**.

- **Rechercher et remplacer :** Exécute une recherche classique et remplace la fonction selon la valeur fournie pour *rechercher* et *remplacer par*. Vous pouvez aussi utiliser des expressions classiques.

> **NOTE**
> Il y a plusieurs implémentations des expressions classiques utilisées. utilise la syntaxe .NET Framework pour les expressions classiques. Pour plus d'informations, consulter la Base de Connaissances article.

- **Remplacer les caractères non imprimables avec des espaces :** Remplace tous les caractères de contrôle de la chaîne par un espace (code décimal ASCII 32). Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Supprimer les caractères non imprimables :** Enlève tous les caractères de contrôle de la chaîne. Les caractères non imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux :** Décode les caractères (ou codes de contrôle) qui sont indisponibles sur le clavier, comme le retour chariot ou le passage à la ligne. utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Chariot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter la section Introduire des caractères spéciaux (Codes de Contrôle).
  Cette option convertit les caractères spéciaux de la syntaxe en caractères binaires réels.

Exemple

Quand vous recevez la séquence de données "<CR><LF>", les utilise comme une chaîne complète de 8 caractères. Activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires **CR** (Retour Chariot - code ASCII 13) et **LF** (Passage à la Ligne - code ASCII 10).

- **Rechercher et supprimer tout avant :** Trouve la chaîne de caractères fournie et efface tous les caractères du début des données jusqu'à la chaîne de caractères. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et supprimer tout après :** Trouve la chaîne de caractères fournie et efface tous les caractères depuis la chaîne de caractères jusqu'à la fin des données. La chaîne de caractères trouvée peut aussi être effacée.
- **Changement de casse** : Modifie tous les caractères dans vos chaînes en lettres capitales ou minuscules.

Exemple

L'élément **`LIST_ITEM`** est défini comme bloc de données et zone d'affectation.

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

2.5. Paramétrer le nom de l'étiquette et de l'imprimante dans les données entrées

En principe, les filtres extraient les valeurs des données reçues et les envoient aux variables de l'étiquette pour l'impression. Dans ces cas-là, le nom de l'étiquette ou de l'imprimante est codé en dur dans les actions. Par exemple, l'action Ouvrir l'étiquette code en dur le nom de l'étiquette, et l'action Définir l'imprimante code en dur le nom de l'imprimante. Toutefois, les données d'entrée peuvent également fournir des *méta-données*. Ce sont les valeurs utilisées dans le traitement de NiceLabel Automationmais qui ne sont pas imprimées sur l'étiquette, telles que le nom de l'étiquette, le nom de l'imprimante, la quantité d'étiquettes, etc.

Pour utiliser les valeurs des méta-champs dans le processus d'impression, effectuer les opérations suivantes.

1. **Reconfiguration du filtre :** Définir de nouveaux champs pour que les données d'entrée comportent aussi les champs de méta-données.
2. **Définition de la variable :** Définir manuellement les variables sur lesquelles les méta-données sont stockées. Elles n'existent pas sur l'étiquette et ne peuvent pas être importées. Utiliser des noms intuitifs, tels que **`NomÉtiquette`**, **`NomImprimante`** et **`Quantité`**. N'importe quel nom de variable conviendra.
3. **Reconfiguration du mappage :** Configurer manuellement l'action Utiliser le Filtre de Données pour relier les méta-champs aux variables.
4. **Reconfiguration de l'action :** Reconfigurer l'action **Ouvrir l'étiquette** pour ouvrir l'étiquette spécifiée par la variable **`NomÉtiquette`**, et l'action Installer l'imprimante pour utiliser l'imprimante spécifiée par la variable **`NomImprimante`**.

Exemple

Le fichier CSV contient les données de l'étiquette, mais fournit aussi les *méta-données* telles que le nom de l'étiquette, le nom de l'imprimante et la quantité d'étiquettes. Le filtre de texte structuré extrait tous les champs, envoie les valeurs relatives aux variables d'étiquette et utilise les *méta-données* pour configurer les actions Ouvrir l'étiquette, Installer l'imprimante et Imprimer l'étiquette.

```
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
```

3. Comprendre les déclencheurs

INFO NIVEAU DE PRODUIT

Cette fonctionnalité n'est pas entièrement disponible avec chaque niveau des produits NiceLabel Automation.

NiceLabel Automation est une application basée sur les événements qui déclenche l'exécution d'actions suite à des changements liés aux événements surveillés. Utiliser tout déclencheur disponible pour surveiller les changements dans les événements, tels que la dépose d'un fichier dans un certain dossier, des données arrivant sur un socket TCP/IP spécifique, un message HTTP ou autres. L'objectif principal d'un déclencheur est de détecter des changements liés aux événements, récupérer les données fournies par l'événement et exécuter des actions.

La plupart des déclencheurs attendent de manière passive que l'événement surveillé se produise. Il y a deux exceptions. Le **Déclencheur de base de données** est un déclencheur actif qui recherche périodiquement les changements dans la base de données. Le **Déclencheur de port série** peut attendre les connexions entrantes, ou peut requérir activement des données à des intervalles de temps déterminés.

Traitement des déclencheurs

La plupart du temps, le déclencheur reçoit des données qui doivent être imprimées sur les étiquettes. Dès que le déclencheur reçoit les données, les actions sont exécutées dans l'ordre défini, de haut en bas. Les données reçues peuvent contenir des valeurs pour les étiquettes. Toutefois, avant de pouvoir utiliser ces valeurs, il faut les extraire des données reçues et les enregistrer dans les variables. Les filtres définissent les règles d'extraction. Quand ils sont exécutés, les filtres enregistrent les valeurs extraites dans les variables mappées. Dès que les données sont enregistrées dans les variables, des actions vont pouvoir utiliser les variables, comme l'action Imprimer l'étiquette.

Une fois qu'un événement est survenu, les données d'entrée fournies sont enregistrées dans un fichier temporaire situé dans le dossier **`%temp%`** de l'utilisateur du service. La variable interne **`DataFileName`** fait référence à l'emplacement du fichier temporaire. Le fichier est supprimé quand le déclencheur termine son exécution.

Propriétés du déclencheur

Pour configurer un déclencheur, il faut définir la manière de collecter les données et les actions à exécuter. En option, il est aussi permis d'utiliser des variables. La configuration du déclencheur comprend trois sections.

- **Paramètres :** Définit les paramètres principaux du déclencheur. Sélectionner l'événement qui sera surveillé pour activer le déclencheur, ou définir le canal de communication entrante. L'onglet des paramètres vous permet de sélectionner le moteur de programmation du script et les options de sécurité. Les options disponibles dépendent du type de déclencheur. Pour plus d'informations, consulter la section Types de Déclencheurs ci-dessous.
- **Variables :** Cet onglet définit les variables nécessaires dans le déclencheur. Généralement, les variables sont importées du masque d'étiquette, il faut donc les mapper aux champs extraits des données d'entrée. Définir éventuellement des variables à utiliser en interne dans les différentes actions. Elles ne seront pas envoyées à l'étiquette. Pour plus d'informations, consulter l'article Variables.
- **Actions :** Cette section définit les actions à exécuter chaque fois que le déclencheur détecte un changement dans l'événement surveillé. Les actions sont exécutées dans l'ordre, de haut en bas. Pour plus d'informations, voir la section Actions.

Types de déclencheurs

- **Déclencheur de Fichier :** Surveille les changements dans un fichier ou un ensemble de fichiers dans le dossier. Le contenu du fichier peut être analysé dans des filtres et utilisé dans des actions.
- **Déclencheur sur port série :** Surveille la communication entrant sur le port série RS232. Le contenu du flux entrant peut être analysé dans des filtres et utilisé dans des actions. Les données peuvent également provenir d'un périphérique externe à intervalles de temps définis.
- **Déclencheur de base de données :** Surveille les changements d'enregistrements dans les tables de la base de données SQL. Le contenu des données retournées peut être analysé et utilisé dans les actions. La base de données est surveillée à intervalles de temps définis. Le déclencheur peut aussi mettre la base de données à jour après avoir exécuté les actions utilisant les requêtes **`INSERT`**, **`UPDATE`** et **`INSERT SQL`**.
- **Déclencheur de planification :** Exécute votre déclencheur à des intervalles de temps définis.
- **Déclencheur serveur TCP/IP :** Surveille le flux de données brutes entrant sur le socket défini. Le contenu du flux entrant peut être analysé par des filtres et utilisé dans des actions. Le déclencheur de Serveur TCP/IP peut être bidirectionnel et utilisé pour fournir des retours d'informations.
- **Déclencheur client TCP/IP :** Transforme votre Automation en un client d'écoute qui se connecte aux serveurs TCP/IP.
- **Déclencheur serveur HTTP :** Surveille le flux de données au format HTTP arrivant sur le socket défini. Le contenu du flux entrant peut être analysé par des filtres et utilisé dans des actions. L'authentification d'utilisateur peut être activée. Il peut être bidirectionnel, fournissant un retour d'informations.
- **Déclencheur Web Service :** Surveille le flux de données entrant sur le Web Service défini. Le contenu du flux entrant peut être analysé dans des filtres et utilisé dans des actions. Il peut être bidirectionnel, fournissant un retour d'informations.
- **Déclencheur de Cloud :** Capture les données de NiceLabel Cloud.

Traitement des Erreurs dans les Déclencheurs

- **Erreurs de configuration :** Le déclencheur a un statut d'erreur s'il est mal configuré ou incomplet. Par exemple, le déclencheur fichier est configuré mais sans précision du nom de fichier à surveiller. Ou l'action pour imprimer des étiquettes est définie mais sans précision du nom de l'étiquette. il faut enregistrer les déclencheurs qui contiennent des erreurs mais pas les lancer dans Automation Manager tant que le problème n'est pas résolu. L'erreur signalée au niveau inférieur de la configuration se propage jusqu'au niveau le plus haut, ce qui rend la localisation de l'erreur plus facile.

> **Exemple**
>
> Si l'une des actions est en statut d'erreur, toutes les actions de niveau supérieur indiquent le statut d'erreur. L'icône d'erreur est affichée dans l'onglet Actions et dans le nom du déclencheur.

- **Configurations croisées :** Il est possible de configurer des déclencheurs qui surveillent le même événement, tels que le même nom de fichier ou l'écoute sur le même port TCP/IP, mais ces déclencheurs ne peuvent pas fonctionner simultanément. Un déclencheur dans Automation Manager ne démarrera que si aucun autre déclencheur de cette configuration ou d'une autre ne contrôle le même événement.

Retour d'informations sur le travail d'impression

Voir la section Retour d'informations sur le travail d'impression.

3.1. Déclencheur de Fichier

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

Le déclencheur de fichier survient si :

- le fichier surveillé change
- l'ensemble de fichiers dans le dossier surveillé change
- un nouveau fichier apparaît dans le dossier surveillé

Selon la configuration du déclencheur, soit Windows alerte le déclencheur qu'un fichier a changé, soit le déclencheur dispose d'une liste horodatée des derniers fichiers écrits et se déclenche quand le fichier reçoit un nouvel horodatage.

NOTE

Utilisation typique : Le système utilisé exécute une transaction qui va générer un fichier déclencheur dans le dossier partagé. Le contenu des données peut être structuré en format CSV, XML etc., ou il peut être structuré dans un ancien format. Dans chaque cas, NiceLabel Automation va lire les données, analyser les valeurs en utilisant des filtres et les imprimer sur les étiquettes. Pour plus d'informations concernant l'analyse et l'extraction de données, consulter l'article Comprendre les filtres.

ASTUCE

Pour vous aider à concevoir les configurations avec le déclencheur de fichier, voir les fichiers d'exemples Automation: CSV Composé, CSV Medium, CSV Simple, etc. Vous trouverez les fichiers d'exemples dans **Aide** > **Fichiers d'exemples**.

Général

Cette section permet de configurer les principaux paramètres de ce déclencheur.

- **Nom :** Spécifie le nom unique du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder puis quand vous les exécutez dans Automation Manager.
- **Description :** Vous permet de décrire le rôle de ce déclencheur. Aide les utilisateurs avec une explication courte sur ce que fait le déclencheur.
- **Détecter le fichier spécifié :** Spécifie le chemin et le nom du fichier dont vous surveillez les changements.
- **Détecter un ensemble de fichiers dans le dossier spécifié :** Spécifie le chemin vers le dossier, dont vous pouvez surveiller les changements, et les noms de fichier. Utiliser les caractères génériques standard de Windows * et ?. Certains types de fichiers sont prédéfinis dans les listes déroulantes, mais vous pouvez également introduire vos propres types.

NOTE

Pour surveiller des dossiers en réseau, il faut utiliser la notation UNC de `\\server\share\file`. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées.

- **Détecter automatiquement les changements :** NiceLabel Automation répond aux changements de fichiers dès qu'il est créé ou modifié. Dans ce cas, le système d'exploitation Windows informe le Service NiceLabel Automation du changement. L'utiliser quand le dossier surveillé est situé sur le disque local et également dans certains environnements réseau.
- **Surveiller les changements dans le dossier à intervalles :** NiceLabel Automation Scanne le dossier pour les changements de fichiers à intervalles de temps définis. Dans ce cas, NiceLabel Automation

surveille lui-même le dossier pour des changements de fichiers. La méthode de recherche est plus lente que la détection automatique. L'utiliser quand la détection automatique n'est pas utilisable dans certains environnements.

Exécution

Les options de la section **Accès aux fichiers** spécifient comment l'application accède au fichier déclencheur.

- **Ouvrir le fichier exclusivement :** Ouvre le fichier déclencheur en mode exclusif. Aucune autre application ne peut accéder au fichier en même temps. Il s'agit de l'option de sélection par défaut.
- **Ouvrir le fichier avec les droits de lecture seule :** Ouvre le fichier déclencheur en mode lecture seule.
- **Ouvrir le fichier avec les droits de lecture écriture :** Ouvre le fichier déclencheur en mode lecture seule.
- **Intervalle entre les tentatives d'ouverture du fichier :** Spécifie la période de temps après laquelle NiceLabel Automation tente d'ouvrir le fichier déclencheur. Si l'accès au fichier n'est pas possible au bout de cette période de temps, NiceLabel Automation signale une erreur.

Les options dans la section **Options de surveillance** spécifient les possibilités de surveillance des fichiers.

- **Surveiller la taille du fichier :** Active la détection de changements, non seulement dans l'horodatage, mais aussi dans la longueur du fichier. Les changements d'horodatage du fichier peuvent ne pas être détectés dans certains cas. Ainsi, Automation vérifie également la taille du fichier modifiée et déclenche les actions.
- **Ignorer les fichiers déclencheurs vides :** Si le fichier déclencheur n'a pas de contenu, il est ignoré. Les actions ne s'exécutent pas.
- **Supprimer le fichier déclencheur :** Lorsque le changement dans le fichier déclencheur a été détecté, et que le déclencheur est activé, Automation supprime le fichier. L'activation de cette option permet de nettoyer le dossier des fichiers traités.

NOTE

NiceLabel Automation crée toujours une sauvegarde des données de déclencheur reçues. Le contenu du fichier déclencheur est enregistré en utilisant un nom de fichier unique. C'est important quand le contenu du fichier déclencheur est réutilisable par certaines actions, telle que **Lancer le fichier de commande**. L'emplacement des données de déclencheur sauvegardées est référencé par la variable interne *DataFileName*.

- **Vider le contenu du fichier :** Suite à l'exécution de l'action, le fichier déclencheur est vidé. C'est utile quand des applications tierces joignent des données au fichier déclencheur. Il faut conserver ce fichier pour y joindre les données, sans pour autant imprimer les anciennes données.
- **Suivre les changements quand le déclencheur est inactif :** Le déclencheur est activé lorsque les fichiers changent pendant que le déclencheur est inactif. Si NiceLabel Automation n'est pas déployé dans un environnement de haute disponibilité avec des serveurs de sauvegarde, les fichiers

déclencheurs entrants peuvent se perdre si le serveur est arrêté. Quand NiceLabel Automation est à nouveau en ligne, les fichiers déclencheurs existants peuvent être traités.

- **Nombre d'exécutions d'actions concurrentes** : Précisez votre nombre d'exécutions d'actions concurrentes. Il n'est pas nécessaire d'attendre la fin de l'exécution pour que la suivante démarre. L'ordre de vos actions de traitement reste inchangé pendant que l'exécution de la même action d'une autre thread peut démarrer simultanément.
  Votre nombre maximal d'exécutions d'actions concurrentes dépend également de la performance de votre équipement. Plus d'informations sur Section 7.1, « Traitement parallèle ».

Autre

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

> **INFO NIVEAU DE PRODUIT**
> Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

> **Exemple**
>
> La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.

Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

> **NOTE**
> Il faut activer la prise en charge de l'impression supervisée. Autrement, ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

> **NOTE**
> enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

3.2. Déclencheur sur port série

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

Le déclencheur de port série s'active quand des données sont reçues sur le port série RS232 surveillé.

Utilisation typique : **(1) Remplacement d'imprimante.** Mettre hors service l'imprimante d'étiquettes existante connectée sur le port série. À sa place, NiceLabel Automation va accepter les données, extraire

du flux d'impression reçu les valeurs pour les objets d'étiquettes et créer un travail d'impression pour le nouveau modèle d'imprimante. **(2) Balances.** Les balances procurent les données concernant les objets pesés. NiceLabel Automation extrait les données requises du flux de données reçu et imprime une étiquette. Pour plus d'informations concernant l'analyse et l'extraction de données, consulter l'article Comprendre les filtres.

> **ASTUCE**
> Pour vous aider à concevoir les configurations avec le déclencheur de port série, voir le fichier d'exemple Scanner et imprimer à partir d'Excel Automation. Vous trouverez les fichiers d'exemples dans **Aide** > **Fichiers d'exemples**.

Général

Cette section permet de configurer les principaux paramètres de ce déclencheur.

- **Nom :** Spécifie le nom unique du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder puis quand vous les exécutez dans Automation Manager.
- **Description :** Vous permet de décrire le rôle de ce déclencheur. Aide les utilisateurs avec une explication courte sur ce que fait le déclencheur.
- **Port :** Spécifie le numéro du port série (COM) sur lequel sont reçues les données entrantes. Prendre un port qui n'est pas utilisé par une autre application, ou périphérique, tel que le pilote d'imprimante. Si le port sélectionné est utilisé, il sera impossible de démarrer le déclencheur dans Automation Manager.

Les options de la section **Paramétrage du port** spécifient les paramètres de communication qui doivent correspondre aux paramètres assignés au périphérique port série.

- **Désactiver l'initialisation du port :** Spécifie que l'initialisation du port n'est pas effectuée après le démarrage du déclencheur dans Automation Manager. Cette option est parfois requise pour les ports COM virtuels.

Exécution

- **Utiliser les données d'initialisation :** Spécifie qu'à chaque activation du déclencheur, la chaîne d'initialisation est envoyée au périphérique série. Certains périphériques série doivent être réveillés ou mis en mode veille avant qu'ils puissent procurer les données. Pour plus d'informations concernant la chaîne d'initialisation, consulter le guide d'utilisation du périphérique. Elle peut comporter des caractères binaires. Pour plus d'informations, consulter la section Introduire des caractères spéciaux.
- **Utiliser l'interrogation de données :** Spécifie que le déclencheur questionne activement le périphérique. Le déclencheur va envoyer les commandes fournies dans les champs de **Contenu** à intervalles réguliers spécifiés. Ce champ peut comporter des caractères binaires. Pour plus d'informations, consulter la section Introduire des caractères spéciaux.

Autre

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

> **INFO NIVEAU DE PRODUIT**
> Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

> **Exemple**
>
> La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.
  Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

> **NOTE**
> Il faut activer la prise en charge de l'impression supervisée. Autrement, ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

> **NOTE**
> enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne **`DataFileName`** pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

3.3. Déclencheur de base de données

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

L'événement de déclenchement de base de données survient quand un changement est détecté dans la base de données surveillée. Il peut s'agir de nouveaux enregistrements ou d'enregistrements existants qui ont été mis à jour. Le déclencheur de base de données n'attend pas un événement de changement, tel que l'arrivée de données. En fait, il extrait les données de la base de données à des intervalles de temps définis.

Utilisation typique : Un système existant dans l'entreprise exécute une transaction qui a pour effet de mettre à jour certaines données dans une table de la base de données. NiceLabel Automation va détecter les enregistrements mis à jour ou nouveaux et va imprimer leurs contenus sur les étiquettes.

Général
Cette section permet de configurer les principaux paramètres de ce déclencheur.

- **Nom :** Spécifie le nom unique du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder puis quand vous les exécutez dans Automation Manager.

- **Description :** Vous permet de décrire le rôle de ce déclencheur. Aide les utilisateurs avec une explication courte sur ce que fait le déclencheur.

- **Connexion à la base de données :** Spécifie la chaîne de connexion à la base de données. Cliquer sur **Définir** pour ouvrir la boîte de dialogue Base de données. L'utiliser pour configurer la connexion à la base de données, y compris le type de base de données, le nom de la table et les informations d'identification de l'utilisateur. Il faut pour cela se connecter à une base de données prenant en charge les commandes SQL. Pour cette raison, il n'est pas possible d'utiliser le déclencheur de base de données pour détecter les changements de données dans les fichiers texte CSV ou feuilles de calcul Microsoft Excel.

  > **NOTE**
  > Les détails de configuration dépendent du type de base de données sélectionné. Les options de la boîte de dialogue dépendent des pilotes de base de données utilisés. Pour les détails de configuration, consulter le guide du pilote de la base de données. Pour plus d'informations concernant la connexion à la base de données, lire l'article Accéder aux bases de données.

- **Vérifier la base de données dans des intervalles de temps :** Spécifie l'intervalle de temps pendant lequel la base de données est sondée pour des changements d'enregistrements.

- **Options de détection et avancées :** Ces options permettent d'affiner le mécanisme de détection d'enregistrements. Quand les enregistrements sont récupérés de la base de données, l'onglet Action affiche automatiquement l'action Pour Chaque Enregistrement, dans laquelle on peut relier les champs des tables aux variables des étiquettes.

  Trouver les enregistrements en fonction de la valeur de champ incrémental unique
  Lorsque cette option est activée, le déclencheur surveille le champ numérique auto-incrémental dans la table. NiceLabel Automation mémorise la valeur du champ pour le dernier enregistrement. Lors du sondage suivant, seules les valeurs plus grandes que celles mémorisées sont collectées.

  Pour configurer cette option, sélectionner le nom de la table où les enregistrements résident **`(nom de table)`**, le champ auto-incrémental **`(champ clé)`** et la valeur de départ du champ **`(valeur par défaut du champ clé)`**. En interne, la variable **`KeyField`** (champ clé) est utilisée pour se référer à la valeur actuelle du champ clé.

  > **NOTE**
  > La dernière valeur du champ clé est mémorisée en interne, mais n'est pas remise à jour dans la configuration, donc la valeur pour **`valeur par défaut du champ clé`** ne change pas dans cette boîte de dialogue. La configuration et/ou démarrer/arrêter ce déclencheur peut être chargée en toute sécurité dans Automation Manager tout en conservant la dernière valeur mémorisée. Toutefois, si vous enlevez la configuration de Automation Manager et la remettez, la valeur du dernier champ clé mémorisé sera réinitialisée à la **`valeur par défaut du champ clé`**.

Trouver les enregistrements et les supprimer

Avec cette option sélectionnée, tous les enregistrements sont obtenus à partir de la table puis supprimés. Pour configurer cette option, sélectionner le nom de la table où résident les enregistrements (**`nom de table`**) et spécifier la clé primaire dans la table (**`champs clés`**). Même si Automation vous permet d'avoir une table sans clé primaire, il est toutefois fortement recommandé de d'en définir une. S'il existe une clé, les enregistrements sont effacés un par un au fur et à mesure du traitement des enregistrements dans les actions.

> **AVERTISSEMENT**
> Si la clé primaire n'existe pas, tous les enregistrements collectés dans le déclencheur actuel sont effacés en même temps. C'est bien quand il n'y a pas d'erreurs de traitement des enregistrements. Toutefois, en cas d'erreur de traitement d'un enregistrement, Automation arrête de traiter les autres enregistrements. Comme tous les enregistrements récupérés durant cet intervalle de sondage ont déjà été effacés sans avoir été traités, cela peut causer une perte de données. C'est pourquoi il est conseillé idée d'avoir une clé primaire.

Exemples de Code SQL

> **NOTE**
> Ces requêtes SQL sont en lecture seule et sont fournies uniquement en tant que référence. Pour fournir les requêtes SQL personnalisées, sélectionner la méthode de détection **Trouver et gérer les enregistrements avec une requête SQL personnalisée**.

Table d'exemple :

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

Exemple de mise à jour d'une requête SQL quand la table contient l'index primaire :

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

Le champ **`ID`** dans la table est défini comme index primaire. La construction **`:ID`** dans la clause WHERE contient la valeur du champ ID dans chaque itération. Pour le premier enregistrement, la valeur de **`ID`** est 1, pour le deuxième enregistrement 2, etc. L'utilisation de la variable est spécifiée par les deux points devant le nom de champ dans la requête SQL.

Exemple de mise à jour d'une requête SQL quand la table n'a pas d'index primaire :

```
DELETE FROM [Table]
```

Quand l'index primaire n'est pas défini dans la table, tous les enregistrements sont effacés de la table quand le premier enregistrement a été traité.

Trouver les enregistrements et les mettre à jour

Dans ce cas, tous les enregistrements sont obtenus à partir de la table et mis à jour. Une valeur personnalisée peut être placée dans un champ de la table pour indiquer que 'ces enregistrements ont déjà été imprimés'. Pour configurer cette option, il faut sélectionner le nom de la table où résident les enregistrements **`(nom table)`**, sélectionner le champ à mettre à jour **`(champ de mise à jour)`** et saisir la valeur qui sera enregistrée dans le champ **`(valeur mise à jour)`**. En interne, la variable **`UpdateValue`** est utilisée dans la requête SQL pour référencer la valeur actuelle du champ (**`Valeur de mise à jour`**).

Même si Automation vous permet d'avoir une table sans clé primaire, il est toutefois fortement recommandé d'en définir une. S'il existe une clé, les enregistrements sont mis à jour un par un au fur et à mesure du traitement des enregistrements dans les actions.

AVERTISSEMENT

Si la clé primaire n'existe pas, tous les enregistrements obtenus dans le déclencheur actuel sont mis à jour en même temps. C'est bien quand il n'y a pas d'erreurs de traitement des enregistrements. Mais si une erreur survient pendant le traitement des enregistrements, l'Automation arrête de traiter les autres enregistrements. Comme tous les enregistrements récupérés durant cet intervalle de sondage ont déjà été effacés sans avoir été traités, cela peut causer une perte de données. C'est pourquoi il est conseillé d'avoir une clé primaire.

Exemples de Code SQL

NOTE

Ces requêtes SQL sont en lecture seule et sont fournies uniquement en tant que référence. Pour fournir les requêtes SQL personnalisées, sélectionner la méthode de détection **Trouver et gérer les enregistrements avec une requête SQL personnalisée**.

Table d'exemple :

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

Exemple de mise à jour d'une requête SQL si la table contient l'index primaire :

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

Le champ **`ID`** dans la table est défini comme index primaire. La construction **`:ID`** dans la clause WHERE contient la valeur du champ ID dans chaque itération. Pour le premier enregistrement, la

valeur de **`ID`** est 1, pour le deuxième enregistrement 2, etc. L'utilisation de la variable est spécifiée par les deux points devant le nom de champ dans la requête SQL. Le champ **`UpdateValue`** est défini dans la configuration de déclencheur, dans le champ d'édition **Valeur de mise à jour**.

Exemple de mise à jour d'une requête SQL quand l'index primaire n'est pas défini dans la table :

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

Quand l'index primaire n'est pas défini dans la table, tous les enregistrements de la table sont mis à jour quand le premier enregistrement a été traité.

Trouver et gérer les enregistrements avec une requête SQL personnalisée
Dans ce cas, la création de requêtes SQL pour l'extraction des enregistrements et la mise à jour des champs ne tiennent qu'à vous. Pour configurer cette option, il faut donner une requête SQL personnalisée pour trouver les enregistrements **`(requête de recherche SQL)`** et une requête SQL personnalisée pour mettre à jour l'enregistrement après le traitement **`(mettre à jour la requête SQL)`**. Cliquer sur le bouton **Test** pour exécuter la requête SQL et visualiser le résultat à l'écran.

Utiliser les champs de la table ou les valeurs de variables du déclencheur comme paramètres dans la clause WHERE de la requête SQL. Mettre le caractère deux points (:) devant le nom de champ ou de variable. Ceci signale à NiceLabel Automation d'utiliser la valeur actuelle de ce champ ou variable.

Exemples de Code SQL
Table d'exemple :

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

Exemple de requête de recherche SQL :

Pour obtenir les enregistrements que vous n'avez pas encore imprimé, effectuer les opérations suivantes. Le champ **`AlreadyPrinted`** (déjà imprimé) ne peut pas contenir la valeur **`Y`** et avoir une valeur vide ou NULLE.

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

À partir de la table d'exemple ci-dessus, les enregistrements avec les valeurs d'ID 2 et 3 seront extraits. Le premier enregistrement a déjà été imprimé et sera ignoré.

Exemple de requête de mise à jour SQL :

Pour marquer les enregistrements déjà imprimés par un **`Y`** dans le champ **`AlreadyPrinted`** (déjà imprimé), effectuer les opérations suivantes :

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

Mettre deux points (**`:`**) devant le nom de variable dans la requête SQL pour l'identifier comme une variable. Vous pouvez utiliser n'importe quel champ de la table pour les paramètres de la clause WHERE. Dans l'exemple, nous mettons à jour le champ **`AlreadyPrinted`** (déjà imprimé) seulement pour les enregistrements traités actuellement (la valeur du champ **`ID`** doit être la même que la valeur de l'enregistrement actuel). De façon similaire, on peut référencer les autres champs de l'enregistrement comme **`:ProductID`** ou **`:CodeEAN`**, ou même référencer les variables définies dans le déclencheur de base de données.

Pour effacer l'enregistrement actuel de la table, effectuer les opérations suivantes :

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

Afficher la requête SQL : Élargir cette section pour afficher la requête SQL générée et écrire votre propre requête, si vous avez sélectionné l'option **Trouver et gérer les enregistrements avec une requête SQL personnalisée**.

Aperçu d'exécution SQL

Pour tester l'exécution des requêtes SQL et visualiser les effets, cliquer sur le bouton Test dans la barre d'outils de la zone d'édition SQL. La section d'Aperçu des données s'ouvre dans le panneau de droite. Cliquer sur le bouton **Exécuter** pour démarrer le code SQL. Pour utiliser les valeurs de champs de la table (avec le signe (:) devant le nom de champ) dans la requête SQL, il faudra leur fournir les valeurs de test.

> **NOTE**
> Si l'Aperçu des données s'ouvre et que vous avez ajouté quelques variables au script, cliquer deux fois sur le bouton **Test**. Cette action ferme et rouvre l'Aperçu des données et met à jour la liste des variables dans l'aperçu.

- **Simuler l'exécution :** Spécifie que tous les changements effectués dans la base de données sont ignorés. Les transactions dans la base de données sont annulées, donc aucune mise à jour n'est inscrite dans la base de données.

Exécution

Les options dans Exécution spécifient quand les mises à jour de la base de données s'effectuent. Le type de mise à jour dépend des Options de détection du déclencheur.

- **Avant le traitement des actions :** Spécifie que les enregistrements sont mis à jour avant que les actions définies pour ce déclencheur soient exécutées.
- **Après le traitement des actions :** Spécifie que les enregistrements sont mis à jour après que les actions définies pour ce déclencheur soient exécutées. Généralement, les enregistrements sont mis à jour après leur traitement.

> **NOTE**
> Mais si nécessaire, elles peuvent être mises à jour pendant l'exécution des actions. Pour plus d'informations, voir l'article Exécution des requêtes SQL.

Autre

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

Exemple

La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.
  Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.

- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

> **NOTE**
> Il faut activer la prise en charge de l'impression supervisée. Autrement, ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

> **NOTE**
> enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne **`DataFileName`** pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

3.4. Déclencheur serveur TCP/IP

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

L'événement de déclencheur serveur TCP/IP survient quand les données sont reçues sur le socket surveillé (numéro d'adresse IP et de port).

Utilisation typique : Un système existant exécute une transaction, qui en fait envoie les données au serveur NiceLabel Automation sur un socket spécifique. Le contenu des données peut être structuré en format CSV, XML etc., ou il peut être structuré dans un ancien format. Dans chaque cas, NiceLabel Automation va lire les données, analyser les valeurs en utilisant des filtres et les imprimer sur les étiquettes. Pour plus d'informations concernant l'analyse et l'extraction de données, consulter l'article Comprendre les filtres.

Général

> **NOTE**
> Ce déclencheur est compatible avec le protocole Internet version 6 (IPv6).

Cette section permet de configurer les principaux paramètres de ce déclencheur.

- **Nom :** Spécifie le nom unique du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder puis quand vous les exécutez dans Automation Manager.
- **Description :** Vous permet de décrire le rôle de ce déclencheur. Aide les utilisateurs avec une explication courte sur ce que fait le déclencheur.
- **Port :** Spécifie le numéro de port sur lequel les données entrantes seront acceptées. Utiliser un numéro de port qui n'est pas utilisé par une autre application. Si le port sélectionné est utilisé, il sera impossible de démarrer le déclencheur dans Automation Manager. Pour plus d'informations concernant les problèmes de sécurité, voir l'article Sécuriser l'accès aux déclencheurs.

> **NOTE**
> S'il y a un hébergement multiple activé sur le serveur (plusieurs adresses IP sur une ou plusieurs cartes réseau), NiceLabel Automation répondra au port défini pour toutes les adresses IP.

- **Nombre maximum de connexions simultanées :** Spécifie le nombre maximum de connexions acceptées. Autant de clients peuvent envoyer des données au déclencheur simultanément.

Les options de la section **Exécution** spécifient quand le déclencheur doit activer et démarrer l'exécution des actions.

- **À la déconnexion du client :** Spécifie que le déclencheur s'active lorsque le client aura envoyé les données et terminé la connexion. C'est un paramètre par défaut.

> **NOTE**
> Ne pas utiliser cette option s'il faut envoyer un rapport d'informations sur l'état du travail d'impression à l'application tierce. Si la connexion reste ouverte, l'envoi d'un rapport peut être fait en utilisant l'action **Envoyer les données au port TCP/IP** avec le paramètre *Répondre à l'expéditeur*.

- **Sur le nombre de caractères reçus :** Spécifie que le déclencheur s'active dès qu'il reçoit le nombre de caractères requis. Dans ce cas, l'application tierce peut garder la connexion ouverte et envoyer continuellement des données. Chaque segment de données doit avoir la même taille.
- **Selon la séquence de caractères reçue :** Spécifie que le déclencheur s'active chaque fois qu'il reçoit le nombre de caractères requis. Utiliser cette option quand la 'fin de données' est toujours identifiée par une chaîne de caractères unique. Cette chaîne peut comporter des caractères spéciaux (binaires) en utilisant le bouton à côté du champ d'édition.
    - **Inclure dans les données du déclencheur :** La séquence de caractères qui détermine l'événement de déclenchement n'est pas coupée des données, mais sera incluse dans les données. Le déclencheur reçoit le flux de données complet.

- **Lorsque rien de nouveau n'a été reçu dans l'intervalle de temps spécifié :** Spécifie que le déclencheur s'active après un temps donné passé depuis le dernier caractère reçu.

Exécution

- **Autoriser les connexions des hôtes suivants :** Spécifie la liste d'adresses IP ou les noms d'hôte des ordinateurs pouvant se connecter au déclencheur. Mettre chaque entrée sur une nouvelle ligne.
- **Refuser les connexions pour les hôtes suivants :** Spécifie la liste d'adresses IP ou les noms d'hôte des ordinateurs qui ne peuvent pas se connecter au déclencheur. Mettre chaque entrée sur une nouvelle ligne.
- **Message de bienvenue :** Spécifie le texte du message qui est renvoyé au client chaque fois qu'il se connecte au déclencheur TCP/IP.
- **Message de réponse :** Spécifie le texte du message qui est renvoyé au client chaque fois que l'action est exécutée. Utiliser cette option quand le client ne se déconnecte pas après la transmission de données et attend la réponse de fin d'exécution de l'action. Si le message de la réponse est codé en dur, il est toujours le même.
- **Encodage du message :** Spécifie le modèle d'encodage des données, de façon à ce que les caractères spéciaux puissent être traités correctement. NiceLabel Automation peut détecter automatiquement l'encodage des données, en fonction de l'entête BOM (fichiers texte), ou de l'attribut d'encodage (fichiers XML).

Autre

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

> **INFO NIVEAU DE PRODUIT**
> Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

Exemple

La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.
  Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

  NOTE
  Il faut activer la prise en charge de l'impression supervisée. Autrement, Automation ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

  NOTE
  Automation enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

Envoi de signaux keepalive (conservation de connexion active)
Votre système de routage réseau peut silencieusement interrompre votre connexion en cas d'absence de trafic TCP/IP pendant quelques minutes. Afin d'éviter d'être déconnecté, vous pouvez choisir d'envoyer périodiquement des signaux keepalive dans NiceLabel Automation.

Activez l'envoi de signaux keepalive dans votre fichier `product.config` :

1. Rechercher le dossier système.
   %PROGRAMDATA%\NiceLabel\NiceLabel 10
2. Faire une copie de sauvegarde du fichier **`product.config`**.
3. Ouvrir **`product.config`** dans un éditeur de texte. Le fichier a une structure XML.
4. Ajouter les lignes suivantes :

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

> **NOTE**
>
> `KeepAliveTime` (durée en millisecondes) : Spécifie combien de temps la socket TCP doit rester inactive avant qu'Automation envoie un signal keepalive et attende que les paquets acquittés soient retournés.
>
> `KeepAliveInterval` (intervalle en millisecondes) : Spécifie à quels intervalles envoyer un autre paquet keepalive lorsque l'hôte ne retourne pas de paquets acquittés.
>
> Lorsque le déclencheur TCP/IP reçoit le paquet keepalive et retourne le paquet acquitté, la minuterie `KeepAliveTime` redémarre.
>
> **Les deux valeurs (`KeepAliveTime` and `KeepAliveInterval`) sont obligatoires pour permettre l'envoi de signaux keepalive et doivent être supérieures à 0.**

5. Enregistrez le fichier `product.config`.
6. Redémarrez votre Automation service.

Les signaux keepalive activés conservent désormais votre connexion active.

3.5. Déclencheur client TCP/IP

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

Le déclencheur client TCP/IP transforme votre Automation en un client d'écoute qui se connecte aux serveurs TCP/IP. Plusieurs appareils et systèmes endossent le rôle d'un serveur TCP/IP : systèmes d'inspection visuelle, imprimantes, automates programmables industriels (PLC), lecteurs, balances, etc. Automation peut s'y connecter et attendre les données entrantes. Après avoir reçu un certain nombre de caractères, une séquence de caractères ou un délai d'expiration, le déclencheur client TCP/IP s'active et commence à exécuter vos actions. Si la connexion échoue, le déclencheur vous permet de vous reconnecter automatiquement.

Utilisation typique : Vous imprimez automatiquement plusieurs types d'étiquettes d'emballages en utilisant une imprimante réseau. Il est crucial que vous sachiez à quel moment votre imprimante a terminé d'imprimer un type d'étiquette avant de commencer à en imprimer un autre. Le déclencheur client TCP/IP vous permet de définir la vérification automatisée permanente de la disponibilité de l'imprimante. Pendant que vous imprimez le premier type d'étiquette, le déclencheur client TCP/IP vérifie le statut de l'impression, et lorsque l'imprimante a terminé, il envoie un message à Automation pour envoyer l'autre type d'étiquette en impression.

Général

- **Nom :** Spécifie le nom unique du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder puis quand vous les exécutez dans Automation Manager.
- **Description :** Vous permet de décrire le rôle de ce déclencheur. Aide les utilisateurs avec une explication courte sur ce que fait le déclencheur.
- **Serveur de destination :** Saisir l'emplacement (adresse IP ou nom de l'hôte) du serveur TCP/IP auquel vous voulez vous connecter.
- **Port :** Spécifie le numéro de port hôte à partir duquel vous allez recevoir les données entrantes. Assurez-vous que les ports de votre pare-feu sont ouverts côté serveur.
- **Intervalle de reconnexion au serveur :** Définit le temps en millisecondes au bout duquel votre Automation tente de se reconnecter à votre serveur TCP/IP.
- **Sur le nombre de caractères reçus :** Spécifie que le déclencheur s'active dès qu'il reçoit le nombre de caractères requis. Dans ce cas, le serveur peut garder la connexion ouverte et envoyer continuellement des données. Chaque segment de données doit avoir la même taille.
- **Selon la séquence de caractères reçue :** Spécifie que le déclencheur s'active chaque fois qu'il reçoit le nombre de caractères requis. Utiliser cette option quand la 'fin de données' est toujours identifiée par une chaîne de caractères unique. Cette chaîne peut comporter des caractères spéciaux (binaires) en utilisant le bouton à côté du champ d'édition.

    - **Inclure dans les données du déclencheur :** La séquence de caractères qui détermine l'événement de déclenchement n'est pas coupée des données, mais sera incluse dans les données. Le déclencheur reçoit le flux de données complet.

- **Si rien de nouveau n'a été reçu dans l'intervalle de temps spécifié :** Spécifie que le déclencheur s'active après un temps donné passé depuis le dernier caractère reçu.

Exécution

- **Message d'initialisation :** Message texte qui arrive sur le serveur dès qu'Automation établit la connexion.
- **Message de réponse :** Spécifie le texte du message qui est renvoyé au serveur à l'activation du déclencheur (avant que les actions ne commencent à être exécutées).
- **Encodage du message :** Spécifie le modèle d'encodage des données, de façon à ce que les caractères spéciaux puissent être traités correctement. NiceLabel Automation peut détecter automatiquement l'encodage des données, en fonction de l'entête BOM (fichiers texte), ou de l'attribut d'encodage (fichiers XML).

Autre

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

> **INFO NIVEAU DE PRODUIT**
> Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

> **Exemple**
>
> La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.
  Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

  > **NOTE**
  > Il faut activer la prise en charge de l'impression supervisée. Autrement, ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

  > **NOTE**
  > enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

Envoi de signaux keepalive (conservation de connexion active)

Votre système de routage réseau peut silencieusement interrompre votre connexion en cas d'absence de trafic TCP/IP pendant quelques minutes. Afin d'éviter d'être déconnecté, vous pouvez choisir d'envoyer périodiquement des signaux keepalive dans NiceLabel Automation.

Activez l'envoi de signaux keepalive dans votre fichier `product.config` :

1. Rechercher le dossier système.
   %PROGRAMDATA%\NiceLabel\NiceLabel 10
2. Faire une copie de sauvegarde du fichier **`product.config`**.
3. Ouvrir **`product.config`** dans un éditeur de texte. Le fichier a une structure XML.
4. Ajouter les lignes suivantes :

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

> **NOTE**
>
> `KeepAliveTime` (durée en millisecondes) : Spécifie combien de temps la socket TCP doit rester inactive avant qu'Automation envoie un signal keepalive et attende que les paquets acquittés soient retournés.
>
> `KeepAliveInterval` (intervalle en millisecondes) : Spécifie à quels intervalles envoyer un autre paquet keepalive lorsque l'hôte ne retourne pas de paquets acquittés.
>
> Lorsque le déclencheur TCP/IP reçoit le paquet keepalive et retourne le paquet acquitté, la minuterie `KeepAliveTime` redémarre.
>
> **Les deux valeurs (`KeepAliveTime` and `KeepAliveInterval`) sont obligatoires pour permettre l'envoi de signaux keepalive et doivent être supérieures à 0.**

5. Enregistrez le fichier `product.config`.
6. Redémarrez votre Automation service.

Les signaux keepalive activés conservent désormais votre connexion active.

3.6. Déclencheur serveur HTTP

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

L'événement de déclenchement du serveur HTTP survient si des données sont reçues sur le socket surveillé (numéro d'adresse IP et de port).

Contrairement au déclencheur TCP/IP, les données reçues ne sont pas un flux de données brutes mais comportent l'entête HTTP normalisée. Les applications tierces doivent utiliser les méthodes POST ou GET. Elles doivent fournir des données dans le corps du message ou la chaîne de requête. Vous pouvez utiliser les deux types de médias dans le corps du message, type MIME ou type Contenu. NiceLabel Automation reçoit le message et extrait les données pertinentes du contenu du message en utilisant un filtre.

Utilisation typique : Le système existant exécute une transaction qui envoie les données au serveur NiceLabel Automation formatées en message HTTP POST sur un socket spécifique. Les données envoyées peuvent être structurées en format CSV, XML etc., ou dans un ancien format. Dans chaque cas, NiceLabel Automation va lire les données, analyser les valeurs en utilisant des filtres et les imprimer sur les étiquettes. Pour plus d'informations concernant l'analyse et l'extraction de données, consulter l'article Comprendre les filtres.

ASTUCE

Pour vous aider à concevoir les configurations du déclencheur de Serveur HTTP, voir le fichier d'exemple Prévisualisation de l'étiquette en tant que réponse HTTP Automation. Vous trouverez les fichiers d'exemples dans **Aide** > **Fichiers d'exemples**.

Fourniture de données

Fournir les données du déclencheur HTTP en utilisant l'une des méthodes suivantes. Les méthodes peuvent être également combinées si nécessaire et utilisées ensemble dans la même requête HTTP.

Données de la chaîne de requête
Une chaîne de requête est la partie de l'URL (localisateur de ressource uniforme) qui contient les données à passer au déclencheur HTTP.

Exemple d'URL typique contenant une chaîne de requête :

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

Le point d'interrogation est utilisé comme séparateur et ne fait pas partie de la chaîne de requête.

La chaîne de requête est composée d'une série de paires **`nom-valeur`**. Dans chaque paire, le nom et la valeur du champ sont séparés par le signe égal (=). Les séries de paires sont séparées par le signe esperluette (&). Une chaîne de requête typique fournit des valeurs pour les champs (variables) dans le format suivant :

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

Le déclencheur HTTP active le support pour l'extraction des valeurs de tous les champs et pour les stocker dans les variables ayant le même nom. Par conséquent, vous n'avez pas à définir de filtre pour extraire les valeurs à partir de la chaîne de requête.

- Il n'y a pas besoin de définir de variables dans le déclencheur pour les remplir avec les valeurs de la chaîne de requête. NiceLabel Automation extrait toutes les variables de la chaîne de requête et envoie leurs valeurs à l'étiquette active. Si des variables de même nom existent dans l'étiquette, leurs valeurs seront remplies avec Automation. Si les variables n'existent pas dans l'étiquette, Automation ignore leurs valeurs sans rapport d'erreurs.
- Si une action requiert des valeurs variables, les définir en faisant correspondre les variables dans le déclencheur. Pour collecter toutes les valeurs fournies par la chaîne de requête, définir simplement des variables ayant le même nom que les champs de la chaîne de requête. Pour l'exemple ci-dessus, il faut définir les variables de déclencheurs avec les noms **`champ1`**, **`champ2`** et **`champ3`**.

Utiliser la méthode de requête GET HTTP pour procurer la chaîne de requête.

Données dans le corps de la requête HTTP
Utiliser la méthode de requête POST pour fournir le message dans le corps de la requête HTTP.

Vous pouvez envoyer n'importe quel type de données ou utiliser n'importe quelle structure de données dans le corps. Vous devez seulement vous assurer que vous pouvez gérer les données en utilisant les filtres NiceLabel Automation. Le contenu peut être formaté en XML, CSV ou texte brut. Le contenu envoyé peut aussi être des données binaires (encodées en Base64). Ne pas oublier d'analyser les données avec des filtres.

Pour influencer la structure du message entrant, utiliser des structures normalisées, telles que XML ou CSV, pour simplifier la configuration du filtre.

Utiliser la méthode de requête POST HTTP pour fournir les données dans le corps du message.

Général

Cette section permet de configurer les principaux paramètres de ce déclencheur.

- **Nom :** Spécifie le nom unique du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder puis quand vous les exécutez dans Automation Manager.
- **Description :** Vous permet de décrire le rôle de ce déclencheur. Aide les utilisateurs avec une explication courte sur ce que fait le déclencheur.

Communication

> **NOTE**
> Ce déclencheur est compatible avec le protocole Internet version 6 (IPv6).

Cette section permet de configurer le numéro du port obligatoire et les options facultatives de retour d'informations. Utiliser les Codes de Réponse standard HTTP pour indiquer le succès de l'action d'exécution. Pour des objectifs plus avancés, vous pouvez aussi renvoyer le contenu personnalisé à l'application fournissant les données. Ce contenu peut être une chaîne de retour d'informations simple ou des données binaires comme la prévisualisation de l'étiquette ou le flux d'impression.

L'URL typique pour se connecter au déclencheur HTTP est la suivante :

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

- **Serveur :** Ceci est l'adresse IP ou FQDN de la machine sur laquelle NiceLabel Automation est installé.
- **Port :** Numéro du port sur lequel sont reçues les données entrantes. Utiliser un numéro de port qui n'est pas utilisé par une autre application. Si le port sélectionné est utilisé, il sera impossible de démarrer le déclencheur dans Automation Manager. Pour plus d'informations concernant les problèmes de sécurité, voir l'article Sécuriser l'accès aux déclencheurs.

  > **NOTE**
  > S'il y a un hébergement multiple activé sur le serveur (plusieurs adresses IP sur une ou plusieurs cartes réseau), NiceLabel Automation répondra au port défini pour toutes les adresses IP.

- **Chemin :** Spécifie le chemin optionnel dans l'URL. Cette fonctionnalité permet au NiceLabel Automation d'exposer plusieurs déclencheurs HTTP sur le même port. Le client utilise les déclencheurs au travers d'un même port dans une syntaxe de type REST, causant l'activation de différents déclencheurs par une URL différente. En cas de doute, utiliser le chemin par défaut (\).

  > **INFO NIVEAU DE PRODUIT**
  > La fonctionnalité de cet élément est disponible dans LMS Enterprise.

- **Connexion sécurisée (HTTPS) :** Elle active la couche de transport sécurisée pour vos messages HTTP et évite l'écoute clandestine. Pour plus d'informations sur la manière de la configurer, voir l'article Utilisation de la couche de transport sécurisée (HTTPS).
- **Chaîne de requête :** Spécifie les paires nom-valeur dans l'URL. C'est un paramètre facultatif. Les données sont habituellement fournies dans le corps de la requête HTTP.
- **Attendre la fin d'exécution du déclencheur :** Le protocole HTTP oblige le destinataire (dans ce cas NiceLabel Automation) à renvoyer une réponse numérique à l'expéditeur en indiquant l'état du message reçu. Par défaut, NiceLabel Automation répond avec le code 200. Cela indique que Automation a bien reçu les données, mais ne donne pas d'informations quant à la réussite des actions de déclenchement.
  Cette option spécifie qu'un déclencheur n'envoie pas de réponse immédiatement après avoir reçu les données, mais attend que toutes les actions aient été exécutées. Après ça, il envoie le code de réponse indiquant la réussite de l'exécution de l'action. Avec cette option est activée, la réponse peut être personnalisée et comporter des données (par ex. : la réponse à une requête HTTP est l'aperçu de l'étiquette en format PDF).
  Voici les codes de réponse HTTP :

Code de réponse HTTP	Description
200	Toutes les actions sont réussies.
401	Non autorisé, erreur dans l'identifiant ou le mot de passe.
500	Erreur au cours de l'exécution de l'action.

> **NOTE**
> Pour envoyer un retour d'informations concernant le processus d'impression, il faut activer le mode d'impression **synchrone**. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

- **Nombre maximum de requêtes simultanées :** Spécifie le nombre maximum de connexions entrantes simultanées. Autant de clients peuvent envoyer des données au déclencheur simultanément. Ce nombre dépend aussi des performances physiques de votre serveur.
  Plus d'informations sur Section 7.1, « Traitement parallèle ».
- **Type de réponse :** Spécifie le type du message de réponse. Les types de médias Internet fréquemment utilisés (connus comme types MIME ou types Content) sont prédéfinis dans le menu déroulant. Si votre type de média n'est pas disponible dans la liste, le saisir simplement vous-même. Automation envoie les données de réponse de sortie en tant que retour d'informations, formatées dans le type de média défini. L'option **Variable** active le type de média variable. Dans ce cas, il faut sélectionner ou créer la variable qui contiendra le type de média.

> **NOTE**
> Si le type de contenu n'est pas spécifié, NiceLabel Automation utilise **`application/octet-stream`** par défaut.

- **Données de réponse :** Définit le contenu du message de réponse. Exemples de réponse HTTP : messages d'erreur personnalisés, aperçu d'étiquette, fichiers PDF, fichier de flux d'impression (fichier spouleur), fichier XML avec les détails du moteur d'impression plus l'aperçu d'étiquette (encodé en chaîne Base64). Les possibilités sont infinies.
  Pour sortir seulement un contenu binaire (tel qu'un aperçu de l'étiquette ou un flux d'impression), il faut sélectionner le type de média approprié, par exemple **`image/jpeg`** ou **`application/octet-stream`**.

- **Entêtes additionnelles** : Permet de définir des entêtes MIME pour le message de réponse HTTP. La syntaxe de l'entête de la réponse et un exemple sont disponibles dans la section Action requête HTTP.

  > **ASTUCE**
  > Dans les données de réponse et les entêtes additionnels, le contenu peut être fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu variable, cliquer sur le bouton avec une flèche à droite de la zone de données et insérer la variable de la liste contenant les données à utiliser. Pour plus d'informations, consulter l'article Utiliser des valeurs composées.

Authentification

- **Aucune :** Aucune méthode d'authentification n'est utilisée.

- **Utilisateur :** Spécifie que les messages entrants comprennent le nom d'utilisateur et mot de passe. Utilisé, le déclencheur acceptera seulement les messages HTTP dont les informations d'identification sont correctes. Pour plus d'informations concernant les problèmes de sécurité, voir l'article Sécuriser l'accès aux déclencheurs.

- **Groupe d'application (défini dans NiceLabel Control Center) :** Comme pour l'authentification **Utilisateur** cette option spécifie aussi que le message entrant doit comporter le nom et le mot de passe de l'utilisateur. Utilisé, le déclencheur acceptera seulement les messages HTTP dont les informations d'identification correspondent à celles des utilisateurs NiceLabel Control Center appartenant à un groupe d'application spécifique.

  - **Groupe** : Plusieurs groupes d'application peuvent être définis sur NiceLabel Control Center. Pour choisir le groupe qui va être autorisé à accéder au déclencheur de Serveur HTTP, utiliser le menu déroulant **Groupe**. Le groupe sélectionné et ses utilisateurs doivent être mis en Actifs quand le déclencheur démarre.

    > **NOTE**
    > Le groupe avec un nom spécifié doit exister sur NiceLabel Control Center lorsque le déclencheur est exécuté. N'importe quel nom de groupe peut être utilisé dans la configuration de Automation Builder. Mais il faut que le nom défini à la fin dans NiceLabel Control Center corresponde à celui qui est défini dans la configuration.

> **ASTUCE**
> Les utilisateurs s'authentifient avec leur identifiant et mot de passe tels que définis dans **NiceLabel Control Center** , onglet **Utilisateurs**. Se référer au Guide utilisateur de NiceLabel Control Center pour plus de détails sur la gestion des utilisateurs (section Utilisateurs et Groupes).

Autre

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

> **INFO NIVEAU DE PRODUIT**
> Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

> **Exemple**
>
> La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.
  Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

> **NOTE**
> Il faut activer la prise en charge de l'impression supervisée. Autrement, ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

> **NOTE**
> enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

3.7. Déclencheur Web Service

> **INFO NIVEAU DE PRODUIT**
> La fonctionnalité de cet élément est disponible dans LMS Enterprise.

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

L'événement de déclencheur Web Service survient si un socket surveillé (adresse IP et numéro de port) reçoit des données. Les données doivent respecter la notation SOAP, données XML encodées dans les

messages HTTP). L'interface du Web Service est décrite dans le document WSDL. Ce document est disponible avec chaque déclencheur Web Service défini.

Le déclencheur Web Service fournit un retour d'informations sur l'état du travail d'impression, mais il faut activer le mode de traitement **synchrone**. Pour plus d'informations, consulter l'article Retour d'informations sur le travail d'impression.

Généralement, les programmeurs utilisent le Web Service pour intégrer l'impression d'étiquettes à leurs propres applications. Un système existant exécute une transaction, qui envoie les données au serveur NiceLabel Automation sur un socket spécifique. Les données envoyées sont formatées en message SOAP. Le contenu de données peut être structuré en format CSV, XML etc., ou il peut être structuré en utilisant l'un des anciens formats. Dans chaque cas, NiceLabel Automation lit les données, analyse les valeurs en utilisant des filtres et les imprime sur les étiquettes. Pour plus d'informations concernant l'analyse et l'extraction de données, consulter l'article Comprendre les filtres.

ASTUCE

Pour vous aider à concevoir les configurations du déclencheur de serveur Web, voir le fichier d'exemple Web Service Automation. Vous trouverez les fichiers d'exemples dans **Aide** > **Fichiers d'exemples**.

Général

Cette section permet de configurer les paramètres généraux de ce déclencheur.

- **Nom :** Spécifie le nom unique du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder puis quand vous les exécutez dans Automation Manager.
- **Description :** Vous permet de décrire le rôle de ce déclencheur. Aide les utilisateurs avec une explication courte sur ce que fait le déclencheur.

Communication

NOTE

Ce déclencheur est compatible avec le protocole Internet version 6 (IPv6).

Cette section permet de configurer le numéro du port obligatoire et les paramètres facultatifs de retour d'informations.

- **Port :** Spécifie le numéro du port qui accepte les données entrantes. Utiliser un numéro de port qui n'est pas utilisé par une autre application. Si le port sélectionné est utilisé, il sera impossible de démarrer le déclencheur dans Automation Manager. Pour plus d'informations concernant les problèmes de sécurité, voir l'article Sécuriser l'accès aux déclencheurs.

> **NOTE**
> S'il y a un hébergement multiple activé sur le serveur (plusieurs adresses IP sur une ou plusieurs cartes réseau), NiceLabel Automation répondra au port défini pour toutes les adresses IP.

- **Connexion sécurisée (HTTPS) :** Elle active la couche de transport sécurisée pour vos messages HTTP et évite l'écoute clandestine. Pour plus d'informations sur la manière de paramétrer une connexion sécurisée, voir l'article Utilisation de la couche de transport sécurisée (HTTPS).
- **Nombre maximum d'appels simultanés :** Spécifie le nombre maximum de connexions acceptées. Autant de clients peuvent envoyer des données au déclencheur simultanément.
- **Données de réponse :** Définit la réponse personnalisée qui peut être utilisée avec les méthodes `ExecuteTriggerWithResponse` et `ExecuteTriggerAndSetVariablesWithResponse`. La réponse est fournie dans la zone de texte. Elle peut comporter des valeurs fixes, des valeurs variables et des caractères spéciaux. Pour insérer (ou créer) des variables et des caractères spéciaux, cliquer sur le bouton flèche à droite de la zone de texte. La réponse peut contenir des données binaires, telles qu'une image d'aperçu de l'étiquette et le fichier d'impression (*.PRN).

Retour d'informations sur l'état

Par sa conception, le déclencheur Web Service renvoie des informations sur les travaux d'impression créés. Le déclencheur acceptera les données fournies et les utilise pour exécuter les actions définies. Vous pouvez superviser l'exécution des actions, le déclencheur renvoie des informations d'état positif pour chaque événement survenant lors de l'exécution. Pour activer le rapport d'état lors du processus d'impression, activer Mode d'impression synchrone.

Le déclencheur Web Service expose les méthodes suivantes (fonctions) :

- **ExecuteTrigger :** (exécuter le déclencheur) Cette méthode accepte l'introduction de données dans le traitement et renvoie en option le rapport d'informations de l'état. L'un des paramètres d'entrée active ou désactive le retour d'informations. Si l'option est activée, le retour d'informations contient l'ID d'erreur et une description détaillée de l'erreur. Si l'ID d'erreur est égale à 0, il n'y a pas eu de problème lors de la création du fichier d'impression. Si l'ID est supérieure à 0, une erreur est survenue lors du processus d'impression. Dans cette méthode, la réponse du Web Service n'est pas configurable, elle contiendra toujours l'ID d'erreur et la description de l'erreur.

- **ExecuteTriggerWithResponse :** (exécuter le déclencheur avec réponse) Cette méthode accepte l'introduction de données dans le traitement et renvoie les informations personnalisées. La réponse de Web Service est configurable. Vous pouvez répondre en utilisant n'importe quel type de données organisées dans n'importe quelle structure disponible. Il est possible d'utiliser des données binaires dans la réponse.
- **ExecuteTriggerAndSetVariables :** Similaire à **ExecuteTrigger** ci-dessus, ce Web Service expose des paramètres entrants supplémentaires qui acceptent les listes de paires formatées *nom-valeur*. Le déclencheur analyse automatiquement la liste, extrait les valeurs et les sauvegarde dans les variables de même nom, inutile donc de créer de filtre d'extraction vous-même.

- **ExecuteTriggerAndSetVariablesWithResponse :** Similaire à **ExecuteTriggerWithResponse** ci-dessus, ce Web Service expose des paramètres entrants supplémentaires qui acceptent les listes de paires formatées *nom-valeur*. Le déclencheur analyse automatiquement la liste, extrait les valeurs et les sauvegarde dans les variables de même nom, inutile donc de créer de filtre d'extraction vous-même.

Pour plus d'informations concernant la structure des messages en utilisant l'une ou l'autre méthode ci-dessus, consulter le chapitre WSDL ci-dessous.

WSDL

Le WSDL (Web Service Description Language) spécifie le style des messages SOAP. Il peut être soit un style de **Remote Procedure Call (RPC)**, soit un style de **document**. Choisir le style que supporte votre application fournissant les données.

Le document WSDL définit les paramètres d'entrée et sortie du Web Service.

Après avoir défini le déclencheur Web Service sur le port 12345, le déployer dans Automation Manager et le lancer. Le WSDL devient disponible sur :

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

Le WSDL présente les différentes méthodes qui fournissent des données au déclencheur. Choisir la méthode la plus appropriée pour ce que vous tentez d'accomplir.

- Les méthodes ayant *WithResponse* (avec réponse) dans leurs noms permettent d'envoyer des réponses personnalisées, tels que des messages d'erreur personnalisés, des aperçus d'étiquette, des fichiers PDF, des fichiers d'impression (*.PRN) et similaires. Les méthodes sans *WithResponse* dans leur nom renverront aussi des informations, mais qui ne sont pas personnalisables. Le rapport contiendra des messages d'erreur par défaut.
- Les méthodes ayant *SetVariables* dans leurs noms vous permettent de fournir une liste de variables dans deux formats prédéfinis. Automation extrait automatiquement les valeurs et les cartographie vers les variables appropriées. Cela vous fait gagner du temps puisqu'il ne faut pas définir de filtre pour extraire et relier les données. Pour les méthodes sans *SetVariables* dans leurs noms, il faut définir le filtre vous-même.

L'interface Web Service définit les méthodes suivantes :

Méthode ExecuteTrigger (exécuter le déclencheur)

La partie principale de la définition est la suivante :

```
<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>
```

Il y a deux variables d'entrée (il faudra fournir les valeurs) :

- **texte :** Le filtre dans la configuration analyse la chaîne d'entrée. En général, la chaîne d'entrée est structurée en tant que fichier CSV ou XML, ce qui facilite l'analyse. Vous pouvez aussi utiliser n'importe quel format de fichier texte.
- **attendre :** C'est un champ bouléen qui spécifie deux choses :
  - Si vous voulez attendre la réponse de l'état du travail d'impression ou non.
  - Si le Web Service doit fournir un retour d'informations ou non.

  En cas de *True* (vrai), utiliser **`1`**. En cas de *False* (faux), utiliser **`0`**. Selon le type de méthode sélectionné, il y a soit une réponse prédéfinie, soit une réponse personnalisée.

Il y a les variables de sortie facultatives suivantes (vous recevez leurs valeurs si vous en faites la demande en plaçant **attendre** sur **`1`**) :

- **ExecuteTriggerResult :** La réponse intégrale contient la valeur 0 en cas de rapport d'erreur(s) de traitement des données. Elle contient un nombre entier supérieur à 0 en cas d'erreurs. L'application qui exécute l'appel Web Service à NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **errorText :** Cette valeur de chaîne contient la réponse de l'état du travail d'impression si une erreur survient lors du traitement du déclencheur.

> **NOTE**
> Si une erreur survient durant le traitement du déclencheur, cet élément est inclus dans le message de réponse XML et sa valeur contient la description de l'erreur. Toutefois, s'il n'y a pas d'erreur, cet élément n'est pas inclus dans la réponse XML.

Méthode ExecuteTriggerWithResponse (exécuter le déclencheur avec réponse)
Cette méthode sera utilisée si le déclencheur envoie la réponse personnalisée après la fin de l'exécution.

Quelques exemples de réponse : messages d'erreur personnalisés, aperçu d'étiquette, fichiers PDF générés, fichier de flux d'impression (fichier spouleur), fichier XML avec les détails du générateur d'impression plus l'aperçu d'étiquette (encodé comme chaîne Base64), les possibilités sont infinies.

La partie principale de la définition est la suivante :

```
<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>
```

Dans l'exemple ci-dessus, il y a deux variables d'entrée (il faudra fournir les valeurs) :

- **texte :** Le filtre dans la configuration analyse la chaîne d'entrée. En général, la chaîne d'entrée est structurée en tant que fichier CSV ou XML, ce qui facilite l'analyse. Vous pouvez aussi utiliser n'importe quel format de fichier texte.
- **attendre :** C'est un champ bouléen qui spécifie deux choses :
  - Si vous voulez attendre la réponse de l'état du travail d'impression ou non.
  - Si le Web Service doit fournir un retour d'informations ou non.

  En cas de *True* (vrai), utiliser **1**. En cas de *False* (faux), utiliser **0**. Selon le type de méthode sélectionné, il y a soit une réponse prédéfinie, soit une réponse personnalisée.

De plus, les variables de sorties facultatives suivantes sont incluses dans l'exemple ci-dessus.

> **NOTE**
> Vous recevez des valeurs de sortie facultatives si vous en faites la demande en plaçant la valeur du champ **attendre** sur **1**.

- **ExecuteTriggerWithResponseResult :** La réponse intégrale contient la valeur 0 s'il n'y a pas de problèmes lors du traitement des données. Cette réponse contient un nombre entier supérieur à 0 en cas d'erreurs. L'application qui exécute l'appel Web Service à NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **responseData :** La réponse personnalisée qui peut être définie dans la configuration du déclencheur Web Service. La réponse est encodée en base 64.
- **errorText :** Si une erreur survient lors du traitement du déclencheur, la chaîne contient la valeur de la réponse de l'état du travail d'impression.

  > **NOTE**
  > S'il y a un rapport d'erreurs lors du traitement du déclencheur, le message de réponse XML inclut l'élément **errorText**. La valeur de cet élément contient la description de l'erreur. Toutefois, s'il n'y a pas d'erreur, cet élément n'est pas inclus dans la réponse XML.

Méthode ExecuteTriggerAndSetVariables (exécuter le déclencheur et définir les variables)
La partie principale de la définition est la suivante :

```
<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>
```

Dans l'exemple ci-dessus, il y a trois variables d'entrée (il faudra fournir les valeurs) :

- **texte :** Le filtre dans la configuration analyse la chaîne d'entrée. En général, la chaîne d'entrée est structurée en tant que fichier CSV ou XML, ce qui facilite l'analyse. Vous pouvez aussi utiliser n'importe quel format de fichier texte.
- **attendre :** C'est un champ bouléen qui spécifie deux choses :
  - Si vous voulez attendre la réponse de l'état du travail d'impression ou non.
  - Si le Web Service doit fournir un retour d'informations ou non.

  En cas de *True* (vrai), utiliser **`1`**. En cas de *False* (faux), utiliser **`0`**. Selon le type de méthode sélectionné, il y a soit une réponse prédéfinie, soit une réponse personnalisée.

- **variableData :** C'est la chaîne de caractères contenant les paires *nom:valeur*. Le déclencheur lit toutes les paires et assigne les valeurs fournies aux variables ayant le même nom. Si la variable n'existe pas dans le déclencheur, le déclencheur élimine la paire *nom:valeur*. Si vous fournissez la liste de variables et leurs valeurs en utilisant cette méthode, vous n'avez pas besoin de définir d'extraction de données avec les filtres. Le déclencheur se charge de faire l'analyse.
  Il y a deux structures disponibles pour le contenu variableData.

  Structure XML
  Le déclencheur fournit les variables dans l'élément racine **`<Variables />`** du fichier XML. Le nom de la variable inclut le nom de l'attribut, et la valeur de la variable inclut la valeur de l'élément.

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

> **NOTE**
> Incorporer les données XML dans la section CDATA. **CDATA**, signifiant **donnée de caractère**, est une section de contenu d'élément qui est marquée pour que l'analyseur l'interprète seulement comme données XML de caractères, et non comme une balise. Par conséquent, le déclencheur gère tout le contenu en tant que donnée de caractère. Par exemple, **`<element>ABC</element>`** est interprété en tant que **`<element>ABC</element>`**. Chaque section CDATA commence par la séquence **`<![CDATA[`** et se termine par la séquence **`]]>`**. Pour résumer, placer simplement les données XML entre ces deux séquences.

Paires nom-valeur

Le déclencheur fournit les variables en utilisant un flux textuel. Chaque paire *nom:valeur* a sa propre ligne. Le nom de variable se situe à gauche du signe égal (=), la valeur de variable se situe à droite.

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

Il y a trois variables de sortie facultatives :

> **NOTE**
> Vous recevez des variables facultatives si vous en faites la demande en plaçant **attendre** sur **1** :

- **ExecuteTriggerAndSetVariablesResult :** La réponse intégrale contient la valeur 0 s'il n'y a pas de problèmes lors du traitement des données. Elle contient un nombre entier supérieur à 0 si le traitement des erreurs signale une ou plusieurs erreurs. L'application qui exécute l'appel Web Service à NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **errorText :** Cette valeur de chaîne contient la valeur de la réponse de l'état du travail d'impression si elle signale une erreur de traitement du déclencheur.

  > **NOTE**
  > En cas d'erreur de traitement du déclencheur, cet élément est inclus dans le message de réponse XML. Sa valeur contient la description de l'erreur. Toutefois, s'il n'y a pas d'erreur, cet élément n'est pas inclus dans la réponse XML.

Méthode ExecuteTriggerAndSetVariablesWithResponse (exécuter le déclencheur et définir les variables avec réponse)

Cette méthode sera utilisée si le déclencheur doit envoyer la réponse personnalisée après la fin de l'exécution.

Quelques exemples de réponse : messages d'erreur personnalisés, aperçu d'étiquette, fichiers PDF générés, fichier de flux d'impression (fichier spouleur), fichier XML avec les détails du générateur d'impression plus l'aperçu d'étiquette (encodé comme chaîne Base64), les possibilités sont infinies.

La partie principale de la définition est la suivante :

```
<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>
```

Il y a trois variables d'entrée (il faudra fournir les valeurs) :

- **texte :** Le filtre dans la configuration analyse la chaîne d'entrée. En général, la chaîne d'entrée est structurée en tant que fichier CSV ou XML, ce qui facilite l'analyse. Vous pouvez aussi utiliser n'importe quel format de fichier texte.
- **attendre :** C'est un champ bouléen qui spécifie deux choses :
  - Si vous voulez attendre la réponse de l'état du travail d'impression ou non.
  - Si le Web Service doit fournir un retour d'informations ou non.

  En cas de *True* (vrai), utiliser **`1`**. En cas de *False* (faux), utiliser **`0`**. Selon le type de méthode sélectionné, il y a soit une réponse prédéfinie, soit une réponse personnalisée.

- **variableData :** C'est la chaîne de caractères contenant les paires *nom:valeur*. Le déclencheur lit toutes les paires et assigne les valeurs fournies aux variables ayant le même nom. Si la variable n'existe pas dans le déclencheur, le déclencheur élimine la paire *nom:valeur*. Si vous fournissez la liste de variables et leurs valeurs en utilisant cette méthode, vous n'avez pas besoin de définir d'extraction de données avec les filtres. Le déclencheur se charge de faire l'analyse.
  Il y a deux structures disponibles pour le contenu variableData.

  Structure XML
  Le déclencheur fournit les variables dans l'élément racine **`<Variables />`** du fichier XML. Le nom de la variable inclut le nom de l'attribut, et la valeur de la variable inclut la valeur de l'élément.

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

  > **NOTE**
  > Incorporer les données XML dans la section CDATA. **CDATA**, signifiant **donnée de caractère**, est une section de contenu d'élément qui est marquée pour que l'analyseur l'interprète seulement comme données XML de caractères, et non comme une balise. Par conséquent, le déclencheur gère tout le contenu en tant que donnée de caractère. Par exemple, **`<element>ABC</element>`** est interprété en tant que **`<element>ABC</element>`**. Chaque section CDATA commence par la séquence **`<![CDATA[`** et se termine par la séquence **`]]>`**. Pour résumer, placer simplement les données XML entre ces deux séquences.

Paires nom-valeur

Le déclencheur fournit les variables en utilisant un flux textuel. Chaque paire *nom:valeur* a sa propre ligne. Le nom de variable se situe à gauche du signe égal (=), la valeur de variable se situe à droite.

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

Il y a trois variables de sortie facultatives :

> **NOTE**
> Vous recevez leurs valeurs si vous en faites la demande en plaçant **attendre** sur **1** :

- **ExecuteTriggerAndSetVariablesWithResponseResult :** La réponse intégrale contient la valeur 0 s'il n'y a pas de problèmes lors du traitement des données. Elle contient un nombre entier supérieur à 0 si la réponse signale une ou plusieurs erreurs. L'application qui exécute l'appel Web Service à NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **responseData :** La réponse personnalisée qui peut être définie dans la configuration du déclencheur Web Service. La réponse est encodée en base 64.
- **errorText :** Cette valeur de chaîne contient la valeur de la réponse de l'état du travail d'impression si elle signale une erreur de traitement du déclencheur.

  > **NOTE**
  > En cas d'erreur de traitement du déclencheur, cet élément est inclus dans le message de réponse XML. Sa valeur contient la description de l'erreur. Toutefois, s'il n'y a pas d'erreur, cet élément n'est pas inclus dans la réponse XML.

Autre

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

> **INFO NIVEAU DE PRODUIT**
> Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

Exemple

La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.
  Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

  NOTE
  Il faut activer la prise en charge de l'impression supervisée. Autrement, Automation ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

> **NOTE**
> Automation enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne **`DataFileName`** pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

3.8. Déclencheur de Cloud

> **INFO NIVEAU DE PRODUIT**
> Nécessite un abonnement à NiceLabel Cloud.

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

Pour en savoir plus sur NiceLabel Label Cloud, lire ici.

Le déclencheur de Cloud vous permet d'intégrer votre NiceLabel Cloud ou Control Center sur site à vos systèmes d'entreprise existants exécutés dans des Clouds privés ou des centres de données dédiés. Si un système d'entreprise existant (comme SAP S/4HANA ou Oracle NetSuite) produit une sortie, une API hébergée sur le Cloud vous permet d'envoyer des requêtes HTTP au déclencheur de Cloud.

Le déclencheur de Cloud vous permet d'imprimer des étiquettes en local, dont le contenu provient de systèmes d'informations de type Cloud. Le déclencheur de Cloud exécuté sur un serveur Automation local utilise des méthodes standard pour accéder aux services basés sur le Cloud. De ce fait, vous pouvez déployer une impression en local de manière sûre et efficace.

Le déclencheur de Cloud permet d'intégrer de manière sûre et transparente votre impression d'étiquettes en local via des applications qui communiquent par Internet ouvert.

Comparé au déclencheur serveur HTTP, le déclencheur de Cloud ne vous demande pas d'ouvrir de ports entrants sur votre pare-feu. Le déclencheur de Cloud utilise une API NiceLabel dédiée exécutée dans le Cloud. C'est pourquoi le déclencheur ne nécessite que le port de sortie 443 ou les ports 9350-9354 ouverts. Dans la plupart des cas, ces ports sont déjà ouverts.

En déployant le déclencheur de Cloud, vous avez deux options :

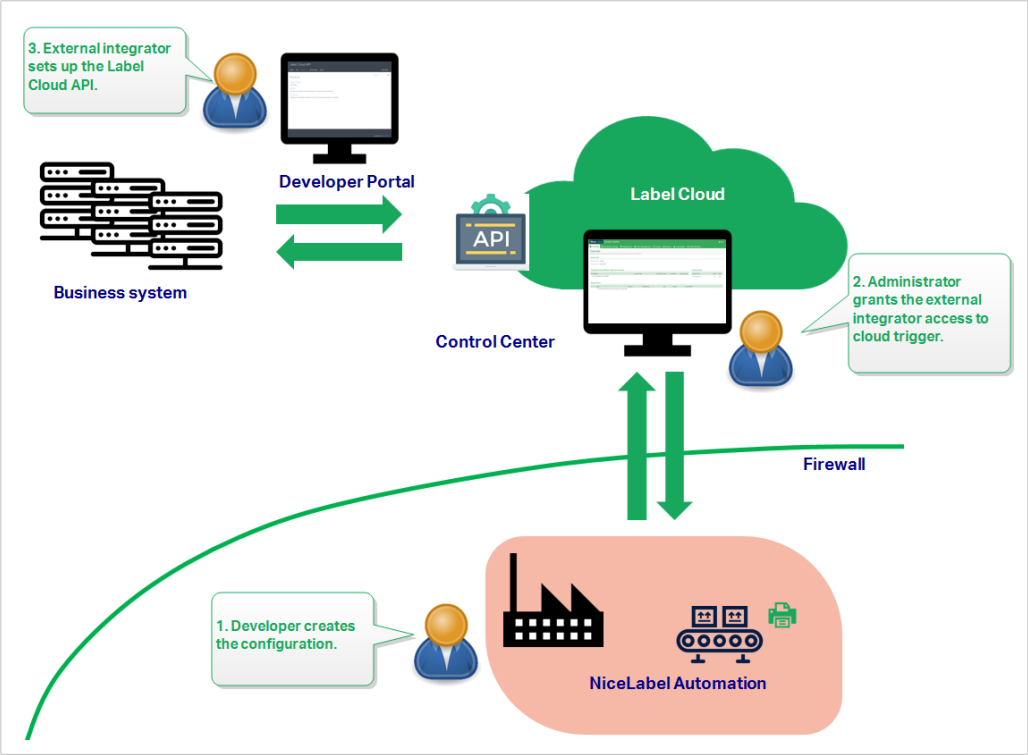
- Vous pouvez déployer le déclencheur dans votre NiceLabel Cloud.
- Vous pouvez déployer le déclencheur dans votre Control Center sur site exécuté en local sur vos serveurs ou dans une infrastructure de Cloud privé.

Les deux options se valent pour ce qui est des fonctionnalités proposées. Choisissez votre option préférée selon votre infrastructure disponible.

3.8.1. Déployer le déclencheur de Cloud avec NiceLabel Cloud

3.8.1.1. Étapes du déploiement pour NiceLabel Cloud

Pour activer l'impression d'étiquettes en local en utilisant le déclencheur de Cloud déployé dans NiceLabel Cloud, vous devez donner trois rôles différents à vos utilisateurs : un utilisateur configure le déclencheur de Cloud sur le serveur local Automation (développeur), un autre paramètre le déclencheur de Cloud dans NiceLabel Cloud, et le troisième gère la souscription sur le Portail développeur.



1. Le **développeur** configure et déploie la configuration du déclencheur de Cloud sur le serveur local Automation en utilisant Automation Builder et Automation Manager.

   > **NOTE**
   > NiceLabel Automation doit être connecté à NiceLabel Cloud.

2. L'administrateur **NiceLabel Cloud** donne à l'intégrateur externe l'accès au déclencheur de Cloud dans Control Center. Ensuite, l'administrateur NiceLabel Cloud envoie la clé d'intégrateur correspondante à l'intégrateur externe.

3. L'**intégrateur externe** rejoint le Portail développeur pour regrouper le système d'entreprise du client et NiceLabel Cloud.

> **NOTE**
> Le Portail développeur héberge l'API dédiée appelée **Déclencheur de Cloud**. L'API sert de point d'interconnexion entre les événements se produisant dans le système d'entreprise du client et la configuration de Automation qui s'exécute en local.

> **NOTE**
> Le mot « externe » signifie que le rôle de cet utilisateur est de s'occuper de la souscription sur le Portail développeur. La souscription créée authentifie l'intégrateur. Les intégrateurs externes ne sont pas forcément des collaborateurs extérieurs. Ils peuvent être des intégrateurs internes appartenant à l'équipe de développement de l'entreprise.

L'intégrateur externe effectue les actions suivantes sur le Portail développeur:

a. Se connecter au Portail développeur. Avant de se connecter pour la première fois, l'intégrateur doit avoir terminé la procédure de souscription.

b. Souscrire à l'API du déclencheur de Cloud.

c. Connecter la souscription à la clé d'intégrateur. C'est ainsi que la souscription donne accès aux déclencheurs de Cloud du client.
Pour plus de détails, voir la section Paramétrer l'API NiceLabel Cloudsur le Portail développeur en détail.

3.8.1.2. Configurer le déclencheur de Cloud dans Automation Builder

Cette section décrit comment configurer le déclencheur de Cloud dans Automation exécuté sur votre serveur local.

1. Ouvrir votre Automation Builder. S'assurer que Automation Builder est connecté à NiceLabel Cloud. Aller dans **Fichier** > **À propos** > Compte **NiceLabel Cloud** pour confirmer votre connexion.

2. L'onglet **Éléments de configuration** s'ouvre. Cliquer sur **Déclencheur de Cloud** pour créer une nouvelle configuration pour le déclencheur de Cloud.

3. Définir un **Nom** et une **Description** pour retrouver facilement votre déclencheur de Cloud parmi les autres déclencheurs.

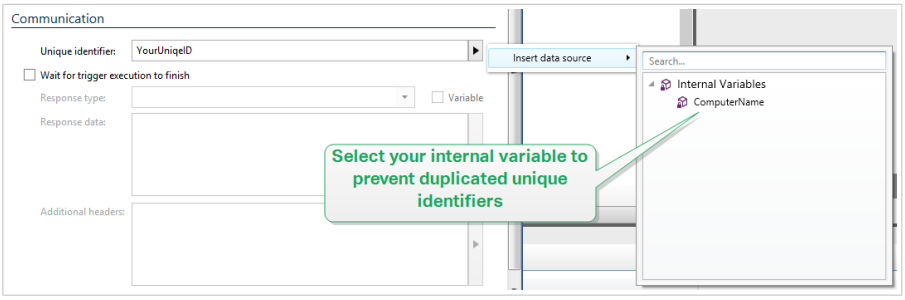
4. Paramétrer le déclencheur **Communication** :

   - Définir l'**identifiant unique**. Après avoir déployé le déclencheur, cet identifiant unique est nécessaire pour appeler le déclencheur.
     Si vous exécutez la configuration du déclencheur de Cloud sur plusieurs ordinateurs, vous devez vous assurer que chaque ordinateur utilise automatiquement son propre identifiant unique. Pour empêcher les doublons indésirables, insérer des variables internes dans l'**identifiant unique**. Vous pouvez utiliser deux variables internes à cette fin :

     - **ComputerName** : Le nom de l'ordinateur sur lequel tourne la configuration.

- **SystemUserName** : Le nom de l'utilisateur actuellement connecté à Windows.

Pour insérer des variables internes dans l'**identifiant unique**, cliquer sur Insérer une source de données et sélectionner vos variables internes.



- **Attendre la fin d'exécution du déclencheur :** Le protocole HTTP oblige le destinataire (dans ce cas NiceLabel Automation) à renvoyer une réponse numérique à l'expéditeur en indiquant l'état du message reçu. Par défaut, NiceLabel Automation répond avec le code 200. Cela indique que Automation a bien reçu les données, mais n'indique pas si les actions du déclencheur ont bien été effectuées.
  Cette option spécifie qu'un déclencheur n'envoie pas de réponse immédiatement après avoir reçu les données, mais attende que les actions se soient toutes exécutées. Ensuite, un code de réponse est envoyé, indiquant que l'action a bien été exécutée. Avec cette option est activée, la réponse peut être personnalisée et comporter des données (par ex. : la réponse à une requête HTTP est l'aperçu de l'étiquette en format PDF).
  Avec le déclencheur de Cloud, les codes de réponse intégrés standard HTTPAutomation sont :

Code de réponse HTTP	Description
200	Toutes les actions sont réussies.
500	Erreur au cours de l'exécution de l'action.

> **NOTE**
> Pour envoyer un rapport d'informations à Automation concernant le processus d'impression, activer le mode d'impression **synchrone**. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

- **Type de réponse :** Spécifie le type du message de réponse. Les types de médias Internet fréquemment utilisés (connus comme types MIME ou types Content) sont prédéfinis dans le menu déroulant. Si votre type de média n'est pas disponible dans la liste, le saisir simplement vous-même. Automation envoie les données de réponse de sortie en tant que retour d'informations, formatées dans le type de média défini. L'option **Variable** active le type de média variable. Dans ce cas, il faut sélectionner ou créer la variable qui contiendra le type de média.

> **NOTE**
> Si le type de contenu n'est pas spécifié, NiceLabel Automation utilise `application/octet-stream` par défaut.

- **Données de réponse :** Définit le contenu de votre message de réponse. Exemples de réponse HTTP : messages d'erreur personnalisés, aperçus d'étiquettes, fichiers PDF, fichiers de flux d'impression (fichier spouleur), fichiers XML avec les détails du moteur d'impression plus l'aperçu d'étiquette (encodé en chaîne Base64).
  Pour sortir seulement un contenu binaire (tel qu'un aperçu de l'étiquette ou un flux d'impression), il faut sélectionner le type de média supporté, par ex. `image/jpeg` ou `application/octet-stream`.

- **Entêtes additionnelles** : Permet de définir des entêtes MIME pour le message de réponse HTTP.
  La syntaxe de l'entête de la réponse et un exemple sont disponibles dans la section Action requête HTTP.

  ASTUCE
  Dans les **données de réponse** et les **Entêtes additionnelles**, le contenu peut être fixe, un mélange de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquer sur le bouton avec une flèche à droite de la zone de données et insérer votre variable de la liste. Vous pouvez aussi créer une nouvelle variable contenant les données que vous voulez utiliser. Pour plus d'informations, consulter l'article Utiliser des valeurs composées.

5. Déployer et lancer le déclencheur dans Automation Manager. Le déclencheur de Cloud surveille maintenant les requêtes entrantes.

NOTE
Si votre configuration nécessite une disponibilité et une évolutivité accrues, vous pouvez déployer plusieurs déclencheurs de Cloud identiques. Pour ce faire, installer plusieurs instances de Automation et y déployer les déclencheurs de Cloud. Si les déclencheurs de Cloud déployés partagent le même **identifiant unique**, la répartition des charges intégrée dans NiceLabel Cloud répartit automatiquement les charges du trafic entre eux.

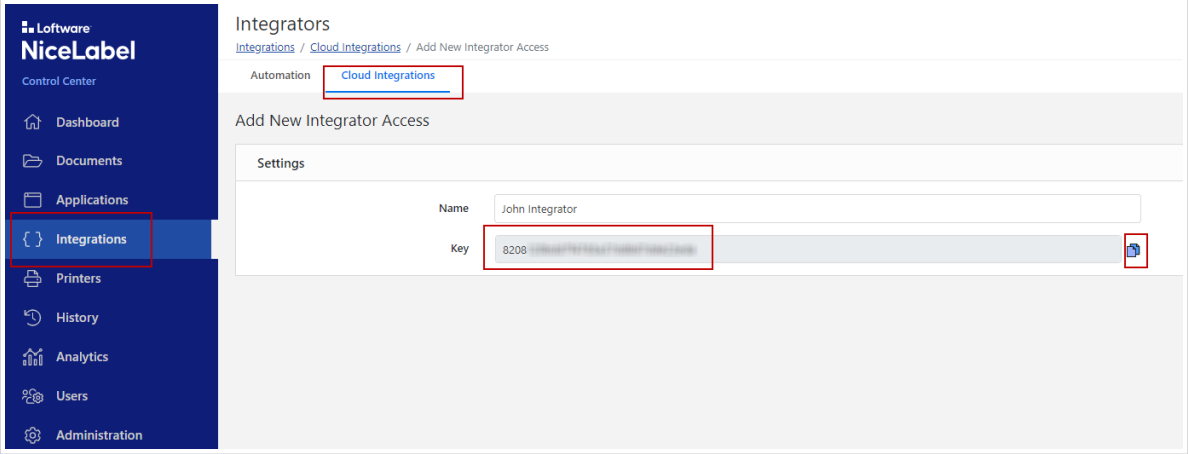
3.8.1.3. Paramétrer l'accès au déclencheur de Cloud pour l'intégrateur externe

NOTE
Pour paramétrer l'accès de l'intégrateur au déclencheur de Cloud, vous devez avoir le **droit de gérer l'intégration au Cloud** sur le Cloud de Control Center. Voir le Guide utilisateur Control Center pour plus de détails concernant la manière dont gérer vos droits utilisateur.

1. Aller dans votre Cloud Control Center. Ouvrir le navigateur Web et saisir :

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

2. Aller dans **Intégrations** > **Intégrations au Cloud**.
3. Cliquer sur **+Ajouter**. Ceci ouvre la page **Ajouter un nouvel accès intégrateur**.
4. Saisir le **Nom** de l'intégrateur que vous êtes en train d'ajouter.



5. Copier la **clé**.
6. Cliquer sur **Enregistrer**.
7. Diriger l'intégrateur externe vers le Portail développeur. Envoyer à l'intégrateur externe les informations suivantes :
   - Lien vers NiceLabel Cloud l'API : https://developerportal.onnicelabel.com/
   - La clé d'intégrateur (voir l'étape 5).
   - L'identifiant unique du déclencheur. Vous trouverez cet **identifiant unique** dans les paramètres de configuration du déclencheur de Cloud Automation (voir étape 4 de la section Configurer le déclencheur de Cloud dans Automation Builder.

> **NOTE**
> L'intégrateur externe doit avoir la clé pour s'authentifier et appeler le déclencheur de Cloud du client.

> **NOTE**
> Pour plus de détails, lire la section Déclencheurs de Cloud dans votre Guide utilisateur Control Center.

3.8.1.4. Créer une souscription sur le Portail développeur

Après avoir reçu les informations requises de l'administrateur NiceLabel Cloud, l'intégrateur externe doit d'abord se connecter au Portail développeur et créer des souscriptions (une par client) pour appeler les déclencheurs. Ces appels de déclencheurs proviennent des systèmes d'informations basés sur le Cloud du client.

NOTE

Si vous ne voyez pas le Portail développeur dans votre messagerie, vérifier dans le Courrier indésirable.

NOTE

Les clients sont des entreprises qui exécutent les configurations de déclencheurs de Cloud qui reçoivent les données des systèmes d'informations externes.

NOTE

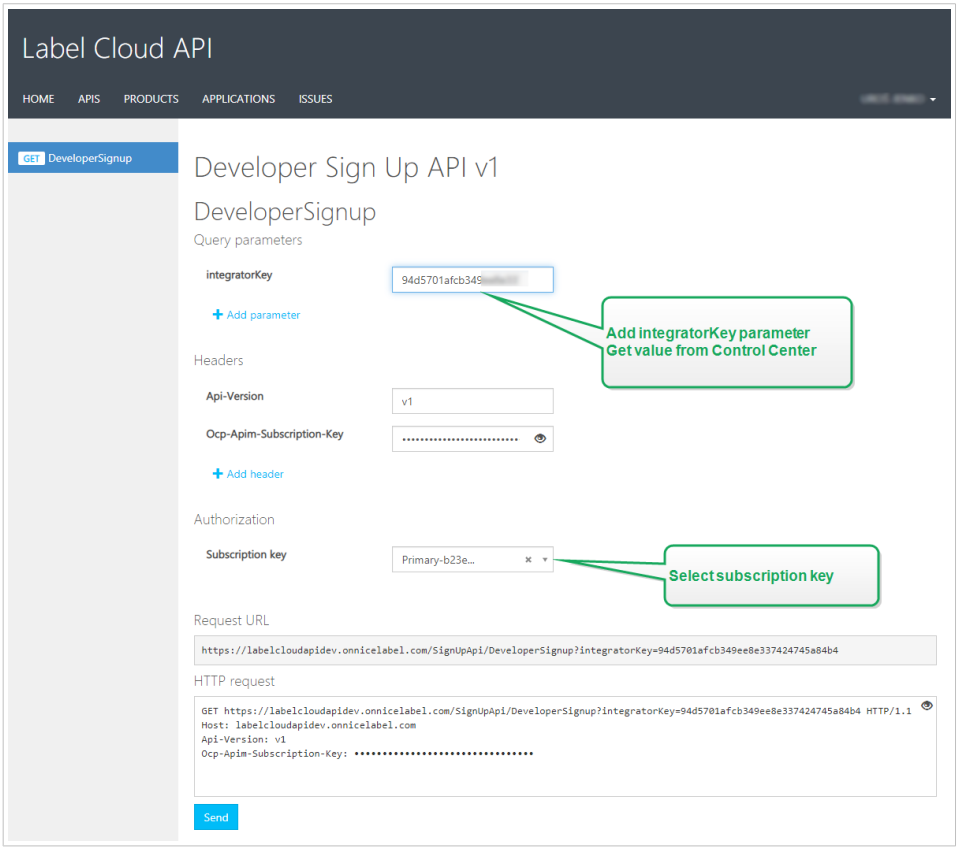
Chaque intégrateur peut appeler plusieurs déclencheurs de Cloud en utilisant une seule souscription.

1. Ouvrir votre navigateur et aller dans https://developerportal.onnicelabel.com/
2. Pour terminer la procédure d'inscription, suivre les instructions à l'écran. Après avoir cliqué sur **Se connecter**, vous recevez un e-mail de confirmation. Cliquer sur le lien de confirmation pour activer votre compte du Portail développeur.
3. Ouvrir l'onglet **Produits** et cliquer sur **Label Cloud**. La page sur laquelle vous arrivez contient vos API et vos souscriptions existantes.
4. Cliquer sur **Ajouter une souscription**. La **Page de souscription au produit** s'ouvre.

   NOTE

   Vous pouvez créer plusieurs souscriptions. Cependant, une seule souscription peut être utilisée par client. C'est pourquoi NiceLabel vous recommande d'inclure le nom de votre client au **nom de la souscription**, tel que `Cloud Trigger Example Customer`.

5. Saisir le **nom de la souscription**.
6. Cliquer sur **Confirmer**. La nouvelle souscription créée devient disponible dans **Produits** > **Label Cloud**.
7. De retour sur la page **Label Cloud**, cliquer sur **Connexion Développeur API v1** (Developer Sign Up API v1).
8. Cliquer sur **Essayer**. La page des API s'ouvre.



9. Insérer la **clé d'intégrateur** depuis le cloud du client Control Center.
10. Cliquer sur **Envoyer**.
    - La réponse est : `Subscription <your subscription key> successfully associated with integrator key <integrator key value>`.

> **NOTE**
> Vous avez reçu la clé d'intégrateur de l'administrateur NiceLabel Cloud. La clé ressemble à ça : `979d7be5df2b473193ac5519f94cd901`

Exemple

En transmettant la clé d'intégrateur en tant que paramètre de requête, l'URL ressemble à ça : `https://labelcloudapi.onnicelabel.com/SignUpApi/DeveloperSignup?integratorKey=979d7be5df2b473193ac5519f94cd901`.

Une fois effectué en utilisant l'URL comme affichée dans l'exemple, l'opération **DeveloperSignup** correspond à la souscription avec le client assigné. Voilà comment l'intégrateur s'authentifie en appelant le déclencheur de Cloud exécuté dans le Automation du client.
Les souscriptions connectées sont également visibles dans le Cloud Control Center. Vérifier si l'intégrateur externe a connecté sa souscription dans**Intégrations** > **Intégrations au Cloud**. L'intégration au Cloud devrait avoir le statut **Développeur [nom, adresse e-mail]** souscrit.

3.8.1.5. Appeler votre déclencheur de Cloud (déploiement de NiceLabel Cloud)

Cette étape s'assure que les sorties des systèmes d'entreprise externes exécutent correctement les déclencheurs de Cloud exécutés en local. C'est le but de l'opération **CloudTrigger** (déclencheur de Cloud). Dans l'URL de l'opération, spécifier le nom du déclencheur que vous appelez.

Pour appeler un déclencheur avec l'identifiant unique `MyCloudTrigger`, utiliser cette URL :

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

Pour chaque événement (sortie) dans le système d'entreprise, appeler l'URL comme dans l'exemple. Chaque appel exécute le déclencheur de Cloud exécuté sur le serveur Automation local.

Toutes les opérations API doivent inclure ces deux entêtes :

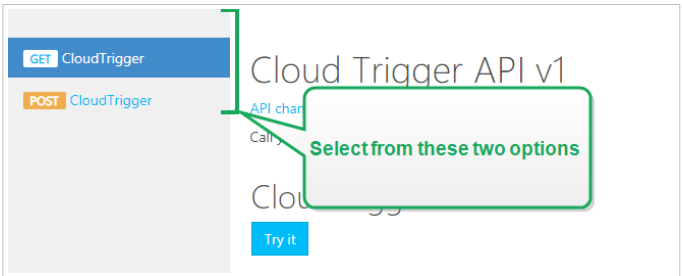
- **Api-Version** est la version de l'API que vous utilisez. Actuellement, la seule version API disponible est v1.
- **Ocp-Apim-Subscription-Key** est la clé qui identifie votre souscription.

Tester les appels CloudTrigger

Pour vous familiariser avec la manière dont fonctionnent les appels **CloudTrigger**, le Portail développeur vous permet de tester ces appels.

Avant de pouvoir tester un appel, vous devez paramétrer une configuration de Automation.

1. Ouvrir le **Portail développeur**, ouvrir l'onglet **Produits** et cliquer sur **Label Cloud**.
2. Sélectionner **Déclencheur de Cloud API v1** (Cloud Trigger API v1).
3. Créer un masque pour les méthodes GET ou POST. Cliquer sur le lien approprié.
   - Après avoir sélectionné la méthode, cliquer sur **Essayer**. Une nouvelle page s'ouvre. Le **triggerID** est déjà ajouté sous **Paramètres de requête**.
   - Dans le champ **Valeur**, copier-coller le **triggerID** du développeur de la configuration d'Automation. C'est l'**identifiant unique** du déclencheur. L'**identifiant unique** est disponible dans **Automation Builder** > **Paramètres des déclencheurs** > **Général**.



4. Sous **Autorisation**, sélectionner la clé de souscription. Étant donné que vous avez déjà au moins une souscription définie, la liste déroulante inclut déjà la clé pour la souscription définie. Sélectionner cette clé, principale ou secondaire.
5. Cliquer sur **Envoyer**.

- Le **statut de la réponse** est 200 OK.

3.8.1.6. Vérifier rapidement si votre déclencheur de Cloud fonctionne

Une fois que vous avez défini votre API NiceLabel Cloud sur le Portail développeur, vous pouvez effectuer une configuration simple dans Automation Builder pour vérifier que le déclencheur de Cloud fonctionne. Si le déclencheur fonctionne, vous recevez le message « Le déclencheur fonctionne » sur la page API du déclencheur de Cloud après avoir cliqué sur **Essayer**.

1. Ouvrir Automation Builder et créer une nouvelle configuration. Assurez-vous que votre Automation Manager est connecté à NiceLabel Cloud.
2. Ajouter un nouveau **Déclencheur de Cloud**.
3. Définir le **Nom** et la **Description**, et définir l'**identifiant unique**. Utilisons l'identifiant unique `TestCloudTrigger` dans ce cas.
4. Activer **Attendre la fin d'exécution du déclencheur**. Cela vous permet de suivre les réponses des déclencheurs.
   - Sélectionner **texte/brut** en tant que **Type de réponse**.
   - Définir les **données de réponse**. C'est ce que vous recevez si le déclencheur fonctionne. Utilisons la chaîne suivante : «Le déclencheur fonctionne.»
   - Définir les **Entêtes additionnelles**. Utiliser le format `Heading:Value`.
5. Déployer la configuration.
6. Ouvrir votre Portail développeur et aller sur la page d'**API du déclencheur Cloud v1**.
7. Cliquer sur **Essayer**.
8. Coller `TestCloudTrigger` au champ **triggerID**. Cliquer sur **Envoyer**.
   - Le contenu de la réponse inclut la confirmation : «Le déclencheur fonctionne.»

3.8.2. Déployer le déclencheur de Cloud avec votre Control Center sur site

3.8.2.1. Configurer le déclencheur de Cloud dans Automation Builder

Cette section décrit comment configurer le déclencheur de Cloud dans Automation exécuté sur votre serveur local.

Ouvrir votre Automation Builder. S'assurer que Automation Builder est associé à votre Control Center. Pour vérifier, aller dans **Fichier** > **Options** > **Control Center** et voir si l'adresse URL de votre Control Center est là.

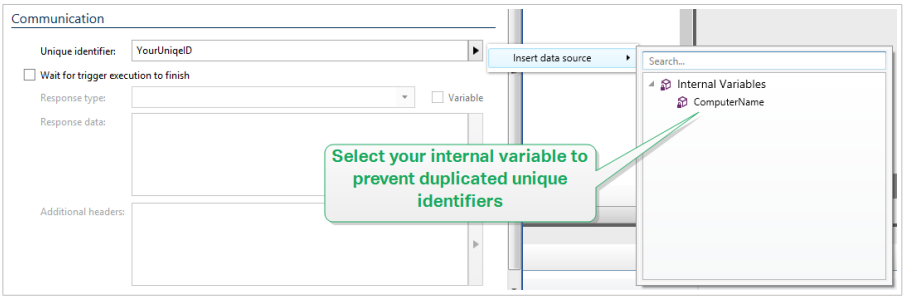
> **NOTE**
> Automation Builder et Control Center « associés » signifient aussi que les deux applications utilisent la même clé de licence.

L'onglet **Éléments de configuration** s'ouvre. Cliquer sur **Déclencheur de Cloud** pour créer une nouvelle configuration pour le déclencheur de Cloud.

Général languette

1. Définir un **Nom** et une **Description** pour retrouver facilement votre déclencheur de Cloud parmi les autres déclencheurs.
2. Paramétrer le déclencheur **Communication** :
   - Définir l'**identifiant unique**. Après avoir déployé le déclencheur, cet identifiant unique enregistre le déclencheur sur votre Control Center. N'utiliser que des caractères alphanumériques. Les caractères spéciaux ne sont pas autorisés.
     Si vous exécutez la configuration du déclencheur de Cloud sur plusieurs ordinateurs, vous devez vous assurer que chaque ordinateur utilise automatiquement son propre identifiant unique. Pour empêcher les doublons indésirables, insérer des variables internes dans l'**identifiant unique**. Vous pouvez utiliser deux variables internes à cette fin :
     - **ComputerName** : Le nom de l'ordinateur sur lequel tourne la configuration.
     - **SystemUserName** : Le nom de l'utilisateur actuellement connecté à Windows.

     Pour insérer des variables internes dans l'**identifiant unique**, cliquer sur Insérer une source de données et sélectionner vos variables internes.



- **Attendre la fin d'exécution du déclencheur :** Le protocole HTTP oblige le destinataire (dans ce cas NiceLabel Automation) à renvoyer une réponse numérique à l'expéditeur en indiquant l'état du message reçu. Par défaut, NiceLabel Automation répond avec le code 200. Cela indique que Automation a bien reçu les données, mais n'indique pas si les actions du déclencheur ont bien été effectuées.
  Cette option spécifie qu'un déclencheur n'envoie pas de réponse immédiatement après avoir reçu les données, mais attende que les actions se soient toutes exécutées. Ensuite, un code de réponse est envoyé, indiquant que l'action a bien été exécutée. Avec cette option est activée, la réponse peut être personnalisée et comporter des données (par ex. : la réponse à une requête HTTP est l'aperçu de l'étiquette en format PDF).
  Avec le déclencheur de Cloud, les codes de réponse intégrés standard HTTPAutomation sont :

Code de réponse HTTP	Description
200	Toutes les actions sont réussies.
400	Aucune configuration disponible.
500	Erreur au cours de l'exécution de l'action.

> **NOTE**
> Pour envoyer un rapport d'informations à Automation concernant le processus d'impression, activer le mode d'impression **synchrone**. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

- **Type de réponse :** Spécifie le type du message de réponse. Les types de médias Internet fréquemment utilisés (connus comme types MIME ou types Content) sont prédéfinis dans le menu déroulant. Si votre type de média n'est pas disponible dans la liste, le saisir simplement vous-même. Automation envoie les données de réponse de sortie en tant que retour d'informations, formatées dans le type de média défini. L'option **Variable** active le type de média variable. Dans ce cas, il faut sélectionner ou créer la variable qui contiendra le type de média.

> **NOTE**
> Si le type de contenu n'est pas spécifié, NiceLabel Automation utilise `application/octet-stream` par défaut.

- **Données de réponse :** Définit le contenu de votre message de réponse. Exemples de réponse HTTP : messages d'erreur personnalisés, aperçus d'étiquettes, fichiers PDF, fichiers de flux d'impression (fichier spouleur), fichiers XML avec les détails du moteur d'impression plus l'aperçu d'étiquette (encodé en chaîne Base64).
  Pour sortir seulement un contenu binaire (tel qu'un aperçu de l'étiquette ou un flux d'impression), il faut sélectionner le type de média supporté, par ex. `image/jpeg` ou `application/octet-stream`.

- **Entêtes additionnelles** : Permet de définir des entêtes MIME pour le message de réponse HTTP.

La syntaxe de l'entête de la réponse et un exemple sont disponibles dans la section Action requête HTTP.

> **ASTUCE**
> Dans les **données de réponse** et les **Entêtes additionnelles**, le contenu peut être fixe, un mélange de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquer sur le bouton avec une flèche à droite de la zone de données et insérer votre variable de la liste. Vous pouvez aussi créer une nouvelle variable contenant les données que vous voulez utiliser. Pour plus d'informations, consulter l'article Utiliser des valeurs composées.

Autre languette

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

> **Exemple**
>
> La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.

Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

> **NOTE**
> Il faut activer la prise en charge de l'impression supervisée. Autrement, ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

> **NOTE**
> enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité languette

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

Gâchette de démarrage

Déployer et lancer le déclencheur dans Automation Manager. Le déclencheur de Cloud surveille maintenant les requêtes entrantes.

NOTE

Si votre configuration nécessite une disponibilité et une évolutivité accrues, vous pouvez déployer plusieurs déclencheurs de Cloud identiques. Pour ce faire, installer plusieurs instances de Automation et y déployer les déclencheurs de Cloud. Si les déclencheurs de Cloud déployés partagent le même **identifiant unique**, la répartition des charges intégrée dans NiceLabel Cloud répartit automatiquement les charges du trafic entre eux.

3.8.2.2. Appeler le déclencheur de Cloud (déploiement sur site)

Cette étape s'assure que les sorties des systèmes d'entreprise externes exécutent correctement les déclencheurs de Cloud exécutés en local. C'est le but de l'opération **CloudTrigger** (déclencheur de Cloud). Dans l'URL de l'opération, spécifier le nom du déclencheur que vous appelez.

Pour appeler le déclencheur avec l'identifiant unique `MyCloudTrigger`, appeler cette URL :

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

NOTE

L'URL peut commencer par «`http`» ou «`https`» – selon la manière dont vous configurez votre Control Center lors de l'installation. Voir le Guide d'installation Control Center, section Définir le site Web et le stockage.

Pour chaque événement (sortie) dans le système d'entreprise, appeler l'URL comme dans l'exemple. Chaque appel exécute le déclencheur de Cloud exécuté sur le serveur Automation local.

Tous les appels doivent inclure l'entête nommée **Integrator-Key** (clé d'intégrateur).

Exemple

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

3.9. Déclencheur de planification

Pour en savoir plus sur les déclencheurs en général, consulter l'article Comprendre les déclencheurs.

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

Le déclencheur de planification fonctionne comme un minuteur qui démarre l'exécution des actions dans votre configuration après une période de temps. Utiliser le déclencheur de planification pour paramétrer l'exécution automatique des actions horodatées.

Le déclencheur de planification est un déclencheur actif. Cela signifie qu'il n'attend pas un changement d'événement mais commence à exécuter les actions assignées dès la fin des intervalles de temps définis.

Exemple

Utilisation typique : Un système ERP produit 6000 fichiers d'étiquettes pour emballages par jour. Automation imprime les étiquettes et stocke les fichiers d'étiquettes utilisés dans un répertoire dédié. En raison d'un grand nombre de fichiers d'étiquettes utilisés, l'entreprise a besoin de définir un nettoyage automatique de fichiers au-delà de 48 heures. Automation utilise le déclencheur de planification pour supprimer les fichiers d'étiquettes obsolètes.

Le déclencheur de planification prend automatiquement en compte le passage à l'heure d'été (DST). Le déclencheur prend toujours le temps du système actuel en compte :

- Lorsque l'heure d'été commence, le déclencheur s'exécute à la première heure après l'heure du temps défini manquant.

Exemple

L'heure du déclencheur de planification est définie sur 2 h 30. L'horloge passe de 2 h à 3 h. Le déclencheur s'exécute à 3 h, heure d'été.

- Pour le passage à l'heure d'hiver, le déclencheur s'exécute seulement à la première occurrence de l'heure définie.

Exemple

L'heure du déclencheur de planification est définie sur 2 h 30. L'horloge passe de 3 h à 2 h. Le déclencheur s'est déjà exécuté. Il n'active pas les options une nouvelle fois lorsque l'horloge indique 2 h 30, heure d'hiver.

3.9.1. Général

Cette section vous permet de configurer les paramètres les plus importants de ce déclencheur.

- **Nom :** Spécifie le nom unique du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder puis quand vous les exécutez dans Automation Manager.
- **Description :** Vous permet de décrire le rôle de ce déclencheur. Aide les utilisateurs avec une explication courte sur ce que fait le déclencheur.

3.9.2. Récurrences

Utiliser les récurrences pour définir la fréquence à laquelle l'événement du déclencheur de planification se répète.

- **Exécuter le déclencheur :** Définit les intervalles de temps de répétition du déclencheur.
  - **Toutes les (X) secondes/minutes/heures** définit le temps de répétition de l'événement du déclencheur dans les unités de temps disponibles. Définir la longueur de l'intervalle dans **Secondes/Minutes/Heures (X)**.
  - **Tous les jours** répète l'événement du déclencheur tous les jours à l'heure spécifiée. Définir l'heure de la répétition quotidienne dans **Heure :**.
  - **À des jours spécifiques** répète le déclencheur à l'heure définie pour un ou plusieurs jours spécifiques. Définir la répétition en utilisant **Heure :** et les jours.

> **NOTE**
> Définir les valeurs de l'heure en format 24 heures.

Autre

Les options de la section **Commentaires du moteur d'impression** spécifient les paramètres de communication qui vous permettent de recevoir un retour d'informations du moteur d'impression.

> **INFO NIVEAU DE PRODUIT**
> Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

- **Impression supervisée :** Active le mode d'impression synchrone. Utiliser cette option pour renvoyer les informations sur l'état du travail d'impression à une application tierce. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Les options de la section **Traitement de données** permettent de préciser s'il faut couper les données pour les ajuster à la variable, ou ignorer les variables manquantes dans l'étiquette. Par défaut, va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement de valeurs trop longues dans les variables d'étiquettes, ou de paramétrage de valeurs dans des variables inexistantes.

- **Ignorer le contenu variable excessif :** tronque les valeurs des données qui dépassent la longueur de la variable telle que définie dans l'éditeur d'étiquettes pour qu'elles s'adaptent. Cette option s'applique lors du paramétrage de valeurs de variables dans les filtres des fichiers de commande et au paramétrage de valeurs de variables de déclencheurs dans les variables d'étiquette ayant le même nom.

Exemple

La variable de l'étiquette accepte un maximum de 5 caractères. Avec cette option activée, toute valeur plus longue que 5 caractères est tronquée aux 5 premiers caractères. Si la valeur est 1234567, ignore les chiffres 6 et 7.

- **Ignorer les variables d'étiquettes manquantes :** Lors de l'impression de fichiers de commande (tels qu'un fichier JOB), le processus d'impression ignore toutes les variables qui sont :
  - spécifiées dans le fichier de commande (en utilisant la commande SET)
  - non définies sur l'étiquette

  La même chose se produit si vous définissez une zone d'assignation dans un filtre pour extraire toutes les paires nom-valeur, mais votre étiquette contient moins de variables.
  Lorsque vous paramétrez des valeurs dans des variables d'étiquettes inexistantes, signale une erreur. Si cette option est activée, l'impression continue.

Les options dans la section **Script** spécifient les possibilités de script.

- **Langage de script :** Sélectionne le langage de script pour le déclencheur. Toutes les actions **Exécuter le script** d'un même déclencheur utilisent le même langage.

Les options de la section **Enregistrer les données reçues** spécifient les commandes disponibles pour les données reçues par le déclencheur.

- **Enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données reçues par le déclencheur. L'option **Variable** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et le nom du fichier.
- **En cas d'erreur, enregistrer les données reçues par le déclencheur vers le fichier :** Activer cette option pour enregistrer les données dans le déclencheur si une erreur survient lors de l'action d'exécution. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème.

  NOTE
  Il faut activer la prise en charge de l'impression supervisée. Autrement, ne peut pas détecter d'erreurs lors de l'exécution. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

  NOTE
  enregistre les données reçues dans un fichier temporaire. Le fichier temporaire est supprimé immédiatement après la fin de l'exécution du déclencheur. La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables internes.

Sécurité

- **Verrouiller et crypter le déclencheur :** Active la protection du déclencheur. Si vous l'activez, le déclencheur est verrouillé et vous ne pouvez plus le modifier. Cela crypte les actions. Seuls les utilisateurs ayant le mot de passe peuvent déverrouiller le déclencheur et le modifier.

3.10. Test des déclencheurs

Après avoir paramétré les déclencheurs, vous avez effectué la moitié de la configuration. Avant de déployer le déclencheur, il faut convenablement tester les opérations qu'il doit effectuer en fonction des données entrantes et vérifier l'exécution des actions.

Automation Builder Vous permet de tester la configuration pendant que vous le développez. Certaines actions ont des capacités d'auto-test, ce qui permet de se concentrer sur l'exécution de l'action elle-même. Le test de chaque déclencheur peut aussi être fait en lançant la commande Afficher l'aperçu. Toutefois, le test final doit toujours se faire dans un environnement réel, en fournissant des données réelles et en utilisant des vrais déclencheurs. C'est ainsi que vous surveillez l'exécution du déclencheur en utilisant Automation Manager.

Test d'exécution des actions individuelles

Certaines des actions incluent l'option d'aperçu. Elle vous permet de modifier les paramètres d'entrée et de voir le résultat de l'action à l'écran.

- **Utiliser un filtre de données :** L'action permet de voir un aperçu en direct des données analysées. Les règles du filtre sélectionné s'appliquent au fichier de données d'entrée sélectionné. Le résultat s'affiche dans la table. S'il y a des sous-zones ou zones d'assignation, l'aperçu présente chaque niveau de définition du filtre.
- **Exécuter une requête SQL :** L'action vous permet de prévisualiser l'exécution de la requête SQL définie. Vous pouvez surveiller l'ensemble des données résultant de l'instruction SELECT et le nombre de rangées affectées par les requêtes UPDATE, INSERT et DELETE. L'exécution de l'aperçu est une transaction sécurisée, ce qui signifie que toutes les modifications peuvent être annulées. Les paramètres de requête peuvent être modifiés pour voir comment ils influencent le résultat.
- **Service Web :** L'action vous permet d'afficher un aperçu d'exécution de la méthode sélectionnée (fonction) du Web Service. Les paramètres d'entrée peuvent être modifiés pour voir comment ils influencent le résultat.
- **Exécuter un script :** L'action vérifiera s'il y a des erreurs de syntaxe dans le script fourni, et l'exécutera. Les paramètres d'entrée peuvent être modifiés pour voir comment ils influencent l'exécution du script.

Tester l'exécution du déclencheur et afficher un aperçu de l'étiquette à l'écran

Pour tester le déclencheur complet, utiliser la fonction **Exécuter l'aperçu**. Cette fonction est utilisable pour chaque déclencheur, quel que soit son type. Le déclencheur ne s'activera pas suite aux changements de l'événement surveillé. Seul un déclencheur initié dans Automation Manager peut le faire. En fait, le déclencheur va effectuer les actions en fonction des données enregistrées dans un fichier. Assurez-vous d'avoir un fichier contenant les données que le déclencheur va accepter durant le déploiement en temps réel.

Le déclencheur exécute toutes les actions définies, y compris le filtrage de données et l'aperçu de l'étiquette à l'écran. L'aperçu montre le processus d'impression dans tous ses détails. Les étiquettes s'impriment comme l'aperçu avec même composition et contenu. Cela inclut le nombre et le contenu des étiquettes. Cela donnera aussi le nombre de travaux d'impression et d'étiquettes générées dans chaque travail ainsi qu'un aperçu de chaque étiquette. Vous pouvez naviguer d'une étiquette à l'autre dans le travail d'impression sélectionné.

Le panneau Journal rapporte les mêmes informations que celles affichées dans le Automation Manager. Développer les entrées du journal pour afficher tous les détails.

> **NOTE**
> En lançant l'aperçu d'impression, toutes les actions définies pour le déclencheur sélectionné sont activées, pas seulement l'action Imprimer l'étiquette. Attention à l'utilisation d'actions modifiant les données, telles qu'Exécuter la requête SQL ou Web Service, car leur exécution est irréversible.

Pour afficher un aperçu des étiquettes, procéder comme suit :

1. Ouvrir la configuration du déclencheur.
2. Vérifier que la configuration du déclencheur est enregistrée.
3. Cliquer sur le bouton **Afficher l'aperçu** dans le groupe Aperçu du ruban.
4. Sélectionner un fichier de données types que le déclencheur va recevoir.
5. Visualiser le résultat dans un onglet Aperçu.

Tester le déploiement sur un serveur de pré-production

Il est conseillé de déployer la configuration dans Automation Manager sur un serveur de pré-production, avant le déploiement sur le serveur de production. Les tests dans un environnement de pré-production peuvent identifier des problèmes de configuration additionnels qui n'ont pas été détectés lors du test du déclencheur dans Automation Builder uniquement.

La performance de la configuration peut aussi être mise à l'épreuve en ajoutant la charge sur le déclencheur pour voir le résultat. Les tests fournissent d'importantes informations concernant la bande passante disponible et identifient les points faibles. En fonction des conclusions, il est possible ensuite d'implémenter diverses techniques d'optimisation, telles qu'optimiser la conception d'étiquettes pour réduire le flux d'impression et optimiser le flux général de données depuis l'application existante vers NiceLabel Automation.

Différences importantes entre les tests réels du déclencheur et l'aperçu dans Automation Builder
Il ne faut pas se fier uniquement à l'aperçu du déclencheur à l'écran dans Automation Builder, même si c'est une méthode rapide de test du déclencheur. Il peut y avoir des différences d'exécution entre l'aperçu et l'activation réelle du déclencheur avec Windows 64 bits.

Même si la configuration fonctionne dans Automation Builder, il faut l'essayer en temps réel.

- En lançant la commande **Afficher l'aperçu**, la configuration l'exécute dans Automation Builder, ce qui lance toujours l'application en 32 bits. L'aperçu de votre déclencheur dans Automation Builder lancera seulement l'exécution de test sur une plateforme 32 bits.
- En lançant le déclencheur en temps réel, la configuration l'exécute en Service, ce qui fonctionne comme une application 32 bits sur Windows 32 bits, et comme une application 64 bits sur Windows 64 bits. Pour plus d'informations, consulter l'article Fonctionnement en mode service.
- Des problèmes peuvent survenir si le traitement du déclencheur est affecté par des différences de plateformes (32 bits contre 64 bits) :
  - **Accès à la base de données :** Les applications 64 bits ont besoin des pilotes de base de données 64 bits, et les applications 32 bits ont besoin des pilotes 32 bits. Pour lancer la configuration d'Automation Builder et dans le Service, Il faut avoir des pilotes de base de données 32 bits et 64 bits afin d'accéder aux bases de données. Pour plus d'informations, voir la section Accéder aux bases de données.
  - **Syntaxe UNC pour les fichiers réseau :** Le compte de service ne peut pas accéder aux fichiers réseau partagés avec une lettre mappée. Il faut utiliser la syntaxe UNC pour les fichiers réseau. Par exemple, utiliser `\\server\share\files\label.lbl` et pas `G:\files\label.lbl`, où G: est mappé à `\\server\share`. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées.
- Attention, si le Service NiceLabel Automation fonctionne sous un autre compte utilisateur que celui de Automation Builder, il peut y avoir des privilèges de sécurité différents. Si l'étiquette est ouverte dans Automation Builder, le compte utilisateur du Service peut ne pas y avoir accès. Pour utiliser Automation Builder sous le même compte utilisateur que le Service, consulter . Utiliser le même compte utilisateur pour configurer et exécuter les déclencheurs.

3.11. Utilisation de la couche de transport sécurisée (HTTPS)

INFO NIVEAU DE PRODUIT
La fonctionnalité de cet élément est disponible dans LMS Enterprise.

Protéger le trafic des données entrant sur Déclencheur de Serveur HTTPet sur Déclencheur Web Service en activant le support HTTPS. HTTPS sécurise la transmission des messages échangés sur le réseau.

La sécurité de communication utilise les certificats X.509 pour encoder les données circulant entre les éléments. Les informations restent confidentielles car seul le client et NiceLabel Automation peuvent décoder le trafic. Et si un utilisateur effectue une écoute clandestine sur la communication, il ne pourra pas comprendre la signification des messages, car le trafic apparaît comme un flux d'octets aléatoires.

C'est une bonne pratique de sécurité d'encoder la communication dans les cas suivants :

- En cas de travail avec des données sensibles et confidentielles qui ne doivent pas être vues par des tiers.
- Quand le message doit traverser des réseaux hors de tout contrôle. Par exemple, cela arrive quand l'envoi des données à Automation se fait par internet, et pas dans le réseau local.

Activer la couche de transport sécurisée (HTTPS)

Pour activer la couche de transport sécurisée pour votre déclencheur, procéder comme suit.

Dans le système Windows :

1. Récupérer le certificat X.509 de l'éditeur de certificats digitaux (autorité de certificats - CA). Il faut un type de certificat pour 'authentification serveur'.

   > **NOTE**
   > Si vous générez vous-même le certificat, il faut importer le certificat CA dans le magasin de l'Autorité de Confiance, pour que la signature CA puisse être vérifiée avec le certificat du serveur.

2. Installer le certificat X.509 dans le système, sur lequel NiceLabel Automation est installé. Il faut que le certificat soit visible par le compte utilisateur sous lequel fonctionne le service NiceLabel Automation. La bonne pratique consiste à installer le certificat dans le magasin de l'ordinateur local, pas dans le magasin de l'utilisateur actuel. Cela va permettre à NiceLabel Automation d'utiliser le certificat, même s'il ne fonctionne pas avec le compte utilisateur connecté.

   a. Ouvrir une fenêtre d'invite de commande.

   b. Saisir **mmc** et appuyer sur la touche ENTER (assurez-vous de l'exécuter avec les droits d'administrateur).

   c. Dans le menu Fichier, cliquer sur **Ajouter/Supprimer l'alignement**.

   d. Dans la boîte de dialogue **Ajouter un alignement autonome**, sélectionner **Certificats**.

   e. Cliquer sur **Ajouter**.

   f. Dans la boîte de dialogue **Alignement des certificats**, sélectionner **Compte de l'ordinateur** et cliquer sur **Suivant**.

   g. Dans la boîte de dialogue **Sélectionner l'ordinateur**, cliquer sur **Terminer**.

   h. Dans la boîte de dialogue **Ajouter/Supprimer l'alignement**, cliquer sur **OK**.

   i. Dans la fenêtre racine de la console, agrandir **Certificats>Personnel**.

   j. Cliquer à droite sur le dossier Certificats et sélectionner **Toutes tâches>Importer**.

   k. Suivre les instructions pour importer le certificat.

3. Extraire l'empreinte d'un certificat que vous venez d'importer.

   a. Toujours dans MMC, double-cliquer sur le certificat.

   b. Dans la boîte de dialogue **Certificat**, cliquer sur l'onglet **Détails**.

   c. Dans la liste de champs, rechercher et cliquer sur Empreinte numérique.

   d. Copier les caractères hexadécimaux du champ. Enlever les espaces entre les nombres hexadécimaux. Par exemple, l'empreinte a9 09 50 2d d8 2a e4 14 33 e6 f8 38 86 b0 0d 42

77 a3 2a 7b doit être spécifiée comme a909502dd82ae41433e6f83886b00d4277a32a7b dans le code. C'est le **`certhash`** requis à l'étape suivante.

4. Lier le certificat à l'adresse IP et au port sur lequel le déclencheur fonctionne. Cette action active le certificat sur le numéro de port sélectionné.
   Ouvrir l'**Invite de Commande** (Il faut l'utiliser avec les droits d'administrateur) et lancer la commande suivante :

```
netsh http add sslcert
ipport=0.0.0.0:56000 certhash=7866c25377554ca0cb53bcdfd5ee23ce895bdfa2
appid={A6BF8805-1D22-42C2-9D74-3366EA463245}
```

   où :

   - **`ipport`** est la paire adresse IP-port, sur laquelle le déclencheur fonctionne. Laisser l'adresse IP à 0.0.0.0 (ordinateur local), mais changer le numéro de port pour qu'il corresponde au numéro de port dans la configuration du déclencheur.
   - **`certhash`** est l'empreinte (SHA hash) du certificat. C'est une chaîne hexadécimale d'une longueur de 20 octets.
   - **`appid`** est le GUID de l'application propriétaire. Tous les GUID sont utilisables, même celui de l'exemple ci-dessus.

Dans la configuration du déclencheur :

1. Dans le déclencheur HTTP ou Web Service, activer l'option **Connexion Sécurisée (HTTPS)**.
2. Recharger la configuration dans Automation Manager.

Désactiver la couche de transport sécurisée (HTTPS)
Dans le système Windows :

- Délier le certificat de la paire Adresse IP-port. Lancer la commande suivante dans l'Invite de Commande (Il faut l'utiliser avec les droits d'administrateur) :

```
netsh http delete sslcert ipport=0.0.0.0:56000
```

  où :

  - `ipport` est la paire adresse IP-port, sur laquelle le déclencheur fonctionne et à laquelle est lié le certificat.

Dans la configuration du déclencheur :

1. Dans le déclencheur HTTP ou Web Service, désactiver l'option **Connexion Sécurisée (HTTPS)**.
2. Recharger la configuration dans Automation Manager.

3.12. Protéger la configuration du déclencheur de toute modification

La configuration du déclencheur peut être protégée en utilisant deux méthodes.

- **Verrouillage du déclencheur.** Cette méthode verrouille la configuration du déclencheur et la protège par un mot de passe. Personne ne peut modifier le déclencheur sans le mot de passe. Activer l'option **Verrouiller et crypter le déclencheur** dans déclencheur *Paramètres ->Sécurité*.
- **Configuration des autorisations d'accès.** Cette méthode permet de fixer des autorisations d'accès pour les utilisateurs. Elles sont définies dans les Options de NiceLabel Automation . Chaque utilisateur peut appartenir à un groupe et chaque groupe peut se voir assigner un rôle différent. Si le groupe a reçu des droits d'édition, tous les membres du groupe pourront éditer les déclencheurs. Cette méthode requiert l'activation de la connexion d'utilisateur. Vous pouvez utiliser les utilisateurs Windows des groupes locaux ou d'Active Directory, ou vous pouvez définir les utilisateurs NiceLabel Automation. Voir **Droit et accès Utilisateur** dans Configuration.

3.13. Configurer un pare-feu pour des déclencheurs réseau

Un déclencheur réseau est un déclencheur qui utilise le protocole TCP/IP. Dans Automation, ce sont des déclencheurs TCP/IP, HTTP ou WEB Service. Ils fournissent des services réseau et sont liés à une carte réseau, son adresse IP et le numéro de port configuré à cet effet. Après avoir déployé et lancer des déclencheurs réseau dans Automation Manager, ils commencent à écouter le port par lequel le trafic arrive.

Un pare-feu protège les ordinateurs des tentatives de connexion entrante non autorisée. Le NiceLabel Installer vérifie que les flux de communication entrante établis sur tous les ports appartenant au Service Automation sont autorisés dans le pare-feu Windows.

AVERTISSEMENT

Le Service Automation possède les ports configurés pour les déclencheurs TCP/IP mais pas les ports définis pour les déclencheurs HTTP ou Web Service. Ces ports sont liés au processus ID 4 (SYSTEM) et non au processus du Service Automation.

Configurer le pare-feu pour permettre la communication sur les ports configurés pour les déclencheurs HTTP et Web Service. Pour créer une règle de trafic entrant, procéder comme suit:

1. Sur l'ordinateur comportant NiceLabel Automation, dans le menu **Démarrer,** sélectionner **Panneau de configuration**, sélectionner **Système et sécurité**, puis sélectionner **Pare-feu Windows**.
2. Dans le panneau de navigation, sélectionner **Paramètres avancés.**
3. Dans la fenêtre **Pare-feu Windows avec fonctions avancées de sécurité,** dans le panneau de navigation, sélectionner **Règles de trafic entrant** puis dans le panneau Actions, sélectionner **Nouvelle règle.**
4. Sur la page **Type de règle,** sélectionner **Port** puis cliquer sur **suivant**.
5. Sur la page **Protocole et Ports**, sélectionner **Ports locaux spécifiques** et donner le numéro du port sur lequel tourne le déclencheur HTTP ou Web Service.
6. Cliquer sur **Suivant**.
7. Sur la page **Actions,** sélectionner **Autoriser la connexion**, et cliquer sur **suivant**.
8. Sur la page **Profil**, sélectionner les profils et cliquer sur **suivant.**
9. Sur la page **Nom**, donner un nom à la règle et cliquer sur **Terminer**.

Renouveler les étapes pour créer un autre pare-feu.

4. Variables

Les variables sont utilisées comme conteneurs de valeurs de données. Elles sont utilisées pour transférer les valeurs à l'étiquette dans l'action **Print Label**, ou pour utiliser les valeurs dans des actions de manipulations de données. Typiquement, le filtre va extraire les valeurs des flux de données reçues par le déclencheur et les envoyer dans les variables. Pour plus d'informations, voir l'article Comprendre les filtres.

La plupart du temps, il faut envoyer les valeurs des variables au masque d'étiquette et imprimer l'étiquette. Le mécanisme qui envoie les valeurs des variables aux étiquettes utilise le mappage de noms automatique. La valeur de la variable définie dans le déclencheur est envoyée à la variable définie dans l'étiquette qui porte le même nom. Il y a trois façons de définir les variables :

- **Importer les variables à partir du fichier d'étiquette :** Pour le mappage automatique expliqué ci-dessus, il est conseillé d'importer chaque fois les variables de l'étiquette. Cette action vous fait gagner du temps et s'assure que les noms des variables correspondent. La variable importée n'hérite pas seulement du nom de la variable, mais aussi de ses propriétés telles que la longueur et valeur par défaut.
- **Définir les variables manuellement :** En définissant des variables manuellement, il faut faire particulièrement attention à l'utilisation de noms identiques à ceux des variables de l'étiquette. Pour définir manuellement les variables qui n'existent pas dans l'étiquette, il est nécessaire qu'elles soient dans le déclencheur.

  NOTE

  Un exemple de variables serait `LabelName`, `PrinterName`, `Quantity` et autres variables similaires dont il faut mémoriser le nom d'étiquette, le nom d'imprimante, la quantité et autres méta-valeurs assignées par le filtre.

- **Activer les variables internes :** Les valeurs des valeurs internes sont assignées par NiceLabel Automation et sont disponibles en lecture seule. Pour plus d'informations, consulter l'article Variables internes.

ASTUCE

Si la **zone d'affectation** (dans les filtres XML et Texte non-structuré) et la **structure dynamique** (dans filtre de Texte Structuré) sont activées, NiceLabel Automation va extraire les paires **nom:valeur** des données du déclencheur et envoyer automatiquement les valeurs aux variables de même nom qui sont définies dans l'étiquette. Le mappage manuel des variables n'est pas nécessaire.

Propriétés

- **Nom :** Spécifie le nom de la variable. Les noms ne sont pas sensibles à la casse. Il est conseillé de ne pas utiliser d'espaces dans les noms. C'est encore plus important pour les variables utilisées dans des scripts ou conditions d'actions, car il faudrait alors les entourer de crochets.
- **Caractères autorisés :** Spécifie la liste des caractères que la valeur peut avoir. Vous avez le choix entre **Tout** (tous les caractères sont acceptés), Numérique (seuls les chiffres sont acceptés), et Binaire (tous les caractères et codes de contrôle sont acceptés).
- **Limiter la longueur de la variable :** Spécifie le nombre maximal de caractères que la variable peut occuper.
- **Longueur fixe :** Spécifie que la valeur doit occuper exactement le nombre de caractères défini par sa longueur.

  > **NOTE**
  > Il est nécessaire de limiter la longueur de la variable pour certains objets de l'étiquette. Un exemple est le code à barres EAN-13, qui accepte 13 chiffres.

- **Valeur requise :** Spécifie que la variable doit contenir une valeur.
- **Valeur par défaut :** Spécifie une valeur par défaut. Si la variable n'a aucune valeur assignée, la valeur par défaut est toujours utilisée.

4.1. Utiliser des valeurs composées

Dans la configuration du déclencheur, certains objets peuvent avoir des valeurs composées. Le contenu peut être un mélange de valeurs fixes, variables et caractères spéciaux (codes de contrôle). Les objets qui acceptent les valeurs composées sont identifiés par un petit bouton flèche sur leur côté droit. Cliquer sur le bouton flèche pour insérer soit une variable, soit un caractère spécial.

- **Utilisation de valeurs fixes :** Saisir une valeur fixe pour la variable manuellement.

  ```
  This is fixed value.
  ```

- **Utiliser des valeurs fixes et des données de variables :** Vous pouvez définir une valeur composée, composée de valeurs variables et fixes. Les noms de variable doivent être entourés par des crochets **[]**. Saisir les variables manuellement, ou les insérer en cliquant sur le bouton flèche sur la droite. Lors du traitement, les variables fusionneront avec les données fixes et seront utilisées comme contenu. Pour plus d'informations, consulter l'article Conseils et astuces d'utilisation de variables dans les actions.
  Dans ce cas, le contenu fusionne trois variables et les données fixes saisies.

  ```
  [variable1] // This is fixed value [variable2][variable3]
  ```

- **Utiliser des caractères spéciaux :** Vous pouvez ajouter des caractères spéciaux au mélange. Saisir les caractères spéciaux manuellement, ou les insérer. Pour plus d'informations, consulter la section Introduire des caractères spéciaux.
  Dans ce cas, la valeur de **`variable1`** est fusionnée avec des données fixes et des caractères binaires de saut de page.

  ```
  [variable1] Form feed will follow this fixed text <FF>
  ```

4.2. Variables internes

Les variables internes sont prédéfinies par NiceLabel Automation. Leurs valeurs sont assignées automatiquement. Elles ne sont disponibles qu'en mode lecture seule. L'icône avec un symbole de cadenas devant le nom de variable distingue les variables internes des variables définies par l'utilisateur. Elles s'utilisent dans les actions de la même façon que les variables définies par l'utilisateur. Les variables internes liées au déclencheur fonctionnent en tant que variables internes pour chaque déclencheur.

Variable interne	Disponible dans le déclencheur	Description
ActionLastErrorDesc	Tous	Fournit la description de la dernière erreur survenue. Utiliser cette valeur en tant que retour d'informations pour héberger le système et identifier la cause de l'erreur.
ActionLastErrorID	Tous	Fournit l'ID de la dernière erreur. C'est un nombre entier. Quand la valeur est 0, il n'y a pas eu d'erreur. Vous pouvez utiliser cette valeur dans les conditions pour évaluer si une erreur est survenue ou non.
BytesOfReceivedData	TCP/IP	Fournit le nombre d'octets reçus par le déclencheur.
ComputerName	Tous	Fournit le nom de l'ordinateur sur lequel la configuration s'exécute.
ConfigurationFileName	Tous	Fournit le chemin et nom de fichier de la configuration actuelle (fichier .MISX).
ConfigurationFilePath	Tous	Fournit le chemin du fichier de configuration actuel. Voir aussi la description de ConfigurationFileName.
DataFileName	Tous	Fournit le chemin et nom de fichier de la copie de travail des données reçues. Chaque fois que le déclencheur accepte les données, il en fait une copie enregistrée dans le fichier identifié par cette variable.
Base de données	Base de données	Fournit le type de base de données comme configuré dans le déclencheur.

Date	Tous	Fournit la date actuelle dans le format spécifié par le système local, telle que 26.2.2018.
DateDay	Tous	Fournit le numéro actuel du jour du mois, tel que 26.
DateMonth	Tous	Fournit le numéro actuel du mois de l'année, tel que 2.
DateYear	Tous	Fournit le numéro actuel de l'année, tel que 2018.
DefaultPrinterName	Tous	Fournit le nom du pilote d'imprimante qui est défini par défaut.
DriverType	Base de données	Fournit le nom du pilote utilisé pour la connexion à la base de données spécifiée.
Hostname	TCP/IP	Fournit le nom du périphérique/de l'ordinateur qui se connecte au déclencheur.
HttpMethodName	HTTP	Fournit le nom de méthode que l'utilisateur a mis dans la requête HTTP, ex. : GET ou POST.
HttpPath	HTTP	Fournit le chemin défini dans le déclencheur HTTP.
HttpQuery	HTTP	Fournit le contenu de la chaîne de requête telle que reçue par le déclencheur HTTP.
NumberOfRowsReturned	Base de données	Fournit le nombre de lignes que le déclencheur reçoit d'une base de données.
LocalIP	TCP/IP	Fournit l'adresse IP locale sur laquelle le déclencheur répond. C'est utile avec une machine d'hébergement multiple avec plusieurs cartes d'interface réseau (NIC) pour déterminer à quelle adresse IP le client s'est connecté. C'est utile en cas de scénario de remplacement de l'imprimante.
PathDataFileName	Tous	Fournit le chemin dans la variable DataFileName, sans le nom du fichier. Voir aussi la description de DataFileName.
PathTriggerFileName	Fichier	Fournit le chemin dans la variable TriggerFileName, sans le nom du fichier. Voir aussi la description de TriggerFileName.
Port	TCP/IP, HTTP, Service Web	Fournit le numéro de port comme défini dans le déclencheur.
RemoteHttpIp	HTTP	Fournit le nom du périphérique/de l'ordinateur qui se connecte au déclencheur.

RemoteIp	Service Web	Fournit le nom du périphérique/de l'ordinateur qui se connecte au déclencheur.
ShortConfigurationFileName	Tous	Fournit le nom du fichier de configuration, sans le chemin. Voir aussi la description de ConfigurationFileName.
ShortDataFileName	Tous	Fournit le nom de la variable DataFileName, sans le chemin. Voir aussi la description de DataFileName.
ShortTriggerFileName	Fichier	Fournit le nom de la variable TriggerFileName, sans le chemin. Voir aussi la description de TriggerFileName.
SystemUserName	Tous	Fournit le nom Windows de l'utilisateur connecté.
TableName	Base de données	Fournit le nom de la table utilisée dans le déclencheur.
Heure	Tous	Fournit l'heure actuelle dans le format spécifié par le système local, telle que 15:18.
TimeHour	Tous	Fournit la valeur de l'heure actuelle, telle que 15.
TimeMinute	Tous	Fournit la valeur actuelle des minutes, telle que 18.
TimeSecond	Tous	Fournit la valeur de secondes actuelle, telle que 25.
TriggerFileName	Fichier	Fournit le nom du fichier qui a déclenché les actions. C'est utile quand vous surveillez un ensemble de fichiers dans le dossier, pour identifier exactement le fichier qui a déclenché les actions.
TriggerName	Tous	Fournit le nom du déclencheur comme défini par l'utilisateur.
Username	Tous	Fournit le nom de l'utilisateur NiceLabel Automation actuellement connecté. La variable a un contenu si la connexion d'utilisateur est requise.

4.3. Variables globales

Les variables globales sont un type de variables utilisables sur plusieurs étiquettes différentes. Les variables globales sont définies en dehors du fichier d'étiquette et mémorisent la dernière valeur utilisée.

En général, les variables globales sont des compteurs globaux. Ces variables globales fournissent une valeur unique pour chaque étiquette demandant une nouvelle valeur. Le fichier est verrouillé pour que chaque valeur soit unique.

Les variables globales sont définies dans l'éditeur d'étiquettes, NiceLabel Automation ne fait que les utiliser. La source des variables globales est configurable dans la boîte de dialogue **Options** (**Fichier** >**Outils** >**Variables globales**).

Par défaut, NiceLabel Automation est configuré pour utiliser les variables globales de l'ordinateur local. L'emplacement par défaut est :

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

Les variables globales sont définies dans les fichiers **`GLOBAL.TDB`** et **`GLOBALS.TDB.SCH`**.

Dans un environnement multi utilisateurs, vous devez configurer tous les clients pour qu'ils utilisent la même source de variables globales partagée en réseau ou les variables globales sur Control Center.

NOTE

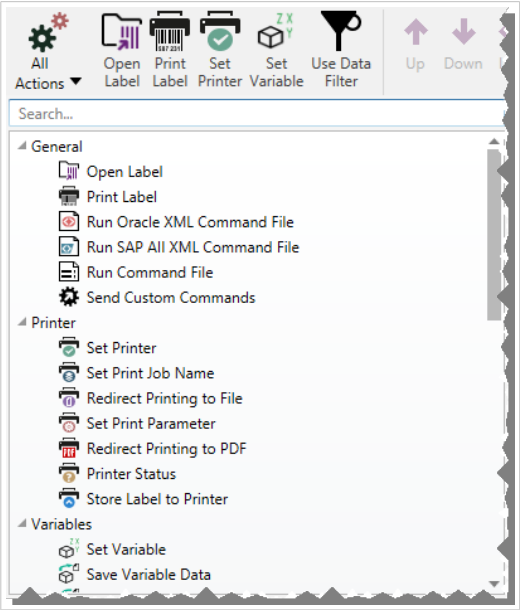
La définition et la valeur des variables globales sont enregistrées dans un fichier ou sur Control Center (pour les produits **LMS Enterprise** et **LMS Pro**).

5. Actions

La section Actions spécifie la liste d'actions qui seront exécutées chaque fois que le déclencheur est activé.

5.1. Définition des actions

Pour définir l'action, cliquez sur l'icône action dans le groupe du ruban Insérer une Action. Le ruban principal contient les actions les plus souvent utilisées. Pour voir toutes les actions disponibles, cliquer sur le bouton **Toutes les Actions**. Pour voir les commandes disponibles pour l'action sélectionnée, cliquer à droite pour sélectionner une commande de la liste.



5.2. Actions indentées

Certaines actions ne peuvent pas être utilisées seules. Leur fonctionnalité spécifique requiert qu'elles soient indentées à une autre action. Utiliser les boutons du groupe de ruban **Ordre d'Action** pour changer la position de l'action. Chaque action est identifiée par le numéro qui montre sa position dans la liste, y compris l'indentation. Ce numéro ID s'affichera dans le message d'erreur, ce qui vous permettra de retrouver plus facilement l'une action problématique.

NOTE

L'action **Imprimer l'Étiquette** est un bon exemple d'une telle action. Il faut la positionner sous l'action **Ouvrir l'Étiquette**, pour qu'elle se réfère à l'étiquette exacte à imprimer.

5.3. Exécution d'Action

La liste des actions ne s'exécute qu'une fois par déclencheur. L'exécution des actions se fait de haut en bas, donc l'ordre des actions est important.

Il y a deux exceptions. Les actions **Boucler** et **Utiliser le filtre de données** exécuteront les actions indentées plusieurs fois. L'action **Boucler** s'exécute aussi souvent que défini dans ses propriétés, et l'action **Utiliser le filtre de données** aussi souvent qu'il y a d'enregistrements dans le bloc de données retourné par le filtre.

NiceLabel 10 s'exécute comme un service sous un compte utilisateur Windows spécifique et hérite les permissions de sécurité du compte. Pour plus d'informations, consulter l'article Fonctionner en Mode Service dans le guide utilisateur de NiceLabel Automation.

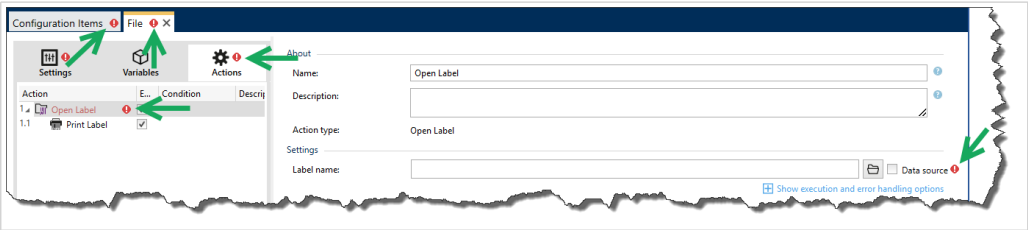
5.4. Actions conditionnelles

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. La condition est une ligne de script (VBScript ou Python). Pour définir la condition, cliquer sur **Afficher les options d'exécution et de traitement d'erreurs** dans les propriétés de l'action pour étendre les possibilités.

Dans ce cas, l'action sera seulement exécutée si l'action précédente s'est terminée avec succès, donc la variable interne `ActionLastErrorID` a la valeur 0. Pour utiliser une telle condition avec des variables internes, il faut d'abord activer la variable interne.

5.5. Identification des actions en état d'erreur de configuration

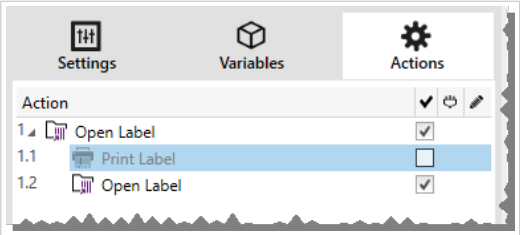
Si l'action n'est pas configurée complètement, elle sera marqué d'un icône avec un point d'exclamation rouge. Une telle action ne peut pas être exécutée. Il est possible d'inclure une telle **Action** dans la liste, mais il faut terminer la configuration avant de pouvoir lancer le déclencheur. Si une des actions indentées est en erreur, toutes les flèches du développement parent (à gauche du nom de l'action) seront également colorées en rouge, pour indiquer l'erreur de sous-action.



Dans ce cas, l'action **Ouvrir l'Étiquette** est en erreur de configuration. Il n'y a pas de paramètre spécifié pour le nom d'étiquette. Le point d'exclamation rouge s'affiche à côté du paramètre erroné dans l'action, dans la liste d'actions, dans l'onglet Actions, dans l'onglet Déclencheur et dans l'onglet Éléments de Configuration, pour faciliter son identification. Le problème est ainsi plus facile à identifier.

5.6. Désactiver les actions

Par défaut, chaque action nouvellement créée est active et sera exécutée au lancement du déclencheur. Désactiver les actions qui ne sont plus nécessaires mais qu'il faut garder dans la configuration. La case à cocher devant le nom de l'action dans la liste d'actions définies est un raccourci pour activer/désactiver l'action.



Dans ce cas, l'action **Imprimer l'étiquette** est encore définie dans la liste d'actions mais a été désactivée. Elle n'est actuellement pas nécessaire et sera ignorée durant l'exécution, mais elle reste facile à réactiver.

5.7. Copier les actions

L'action peut être copiée et collée sur le même déclencheur ou sur un autre. Utiliser les raccourcis-clavier Windows standard ou effectuer un clic-droit sur l'action.

Un clic-droit sur l'action affiche les raccourcis de commandes disponibles pour l'objet sélectionné.

Automation Builder permet aussi de sélectionner plusieurs actions, et de les copier, coller et supprimer. Pour les sélectionner, utiliser les touches Ctrl/Maj plus clic sur les actions requises.

> **NOTE**
> Pour sélectionner plusieurs actions, il faut qu'elles soient dans la même action parent, ex : toutes les actions sélectionnées doivent être au même niveau.

5.8. Naviguer dans la liste d'actions

Utiliser la souris pour sélectionner l'action définie et puis cliquer sur le bouton avec une flèche correspondant dans le groupe **Ordre d'Action** du ruban. Ou utiliser le clavier. Les touches du curseur vont déplacer la sélection dans la liste d'actions, les touches Ctrl + Flèches vont déplacer la position de l'action vers le haut ou le bas, et à gauche ou à droite pour l'indentation.

5.9. Description des actions

Le groupe **A propos** permet de décrire toutes les actions NiceLabel 10.

- **Nom :** par défaut, le nom de l'action est défini par son type; il n'est donc pas unique. En personnalisant son nom, elle devient reconnaissable instantanément au milieu des autres actions, dans les journaux et dans les éventuels messages d'erreur.
- **Description** : Notes de l'utilisateur sur l'action sélectionnée. La description s'affiche dans l'explorateur d'actions.
- **Type d'action** : champ en lecture seule qui affiche le type de l'action.

5.10. Général

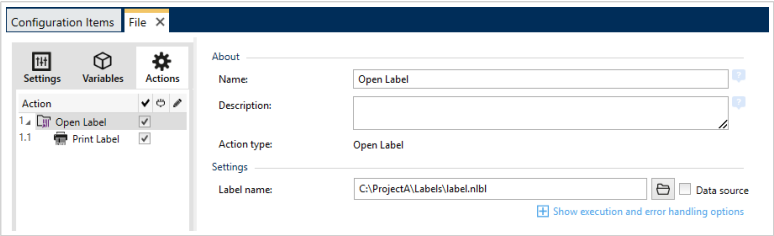
5.10.1. Ouvrir une étiquette

L'action **Ouvrir une étiquette** spécifie le fichier de l'étiquette qui va être imprimée. Quand l'action est exécutée, le masque d'étiquette spécifié s'ouvre en mémoire cache. L'étiquette reste en cache tant que les déclencheurs et les événements l'utilisent.

Le nombre d'étiquettes pouvant être ouvertes en même temps n'est pas limité. Quand une étiquette déjà chargée est à nouveau demandée, NiceLabel Automation va d'abord déterminer si une version plus récente est disponible et approuvée pour l'impression, et ouvrira ensuite l'étiquette.

Dans cet exemple, NiceLabel 10 charge l'étiquette **`label.nlbl`** du dossier **`C:\ProjectA\Labels`**.

```
C:\ProjectA\Labels\label.nlbl
```



S'il ne trouve pas l'étiquette spécifiée, NiceLabel 10 essaye de la trouver à un autre endroit. Pour plus d'informations, consulter l'article Ordre de recherche des fichiers requis dans le guide utilisateur de NiceLabel Designer.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Utilisation de chemins relatifs

NiceLabel 10 permet d'utiliser des chemins relatifs pour référencer les fichiers d'étiquettes. Le dossier racine est toujours le dossier où la solution (ou configuration dans le cas où l'action est utilisée dans une configuration NiceLabel Automation) est stockée.

En utilisant la syntaxe suivante, l'étiquette se chargera de façon relative depuis l'emplacement du fichier de configuration. Automation Buildercherchera l'étiquette dans le dossier **`ProjectA`**, à deux niveaux plus haut que le dossier actuel, et ensuite dans le dossier **`Labels`**.

```
..\..\ProjectA\Labels\label.nlbl
```

Le groupe **Paramètres** définit le fichier de l'étiquette.

- **Nom :** Spécifie le nom de l'étiquette. Il peut être codé en dur, et la même étiquette sera imprimée à chaque fois. L'option **Source de données** permet de définir le nom du fichier dynamiquement. Sélectionner ou créer une variable qui contient le chemin et/ou le nom du fichier, si un déclencheur s'exécute ou un événement survient.

> **ASTUCE**
> En général, la valeur est assignée à la variable par un filtre.

> **NOTE**
> Utiliser la syntaxe UNC pour les ressources réseau.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.10.2. Imprimer l'étiquette

Cette action exécute l'impression de l'étiquette. Elle doit toujours être indentée sous l'action Ouvrir l'étiquette. L'indentation permet d'avoir la référence de l'étiquette à imprimer. Ceci permet d'avoir plusieurs étiquettes ouvertes en même temps, et de spécifier l'étiquette à imprimer.

Avec cette commande, l'étiquette s'imprime en utilisant les pilotes définis dans le masque d'étiquette. Si ce pilote d'imprimante n'existe pas sur le système, l'étiquette s'imprime en utilisant le pilote d'impression par défaut. Il est possible de remplacer le pilote d'imprimante en utilisant la commande Définir l'imprimante.

Pour atteindre une haute performance d'impression d'étiquettes, NiceLabel 10 active deux paramètres par défaut :

- **Traitement parallèle.** De multiples processus d'impression sont effectués simultanément. Le nombre de tâches d'impression fonctionnant en arrière-plan dépend du matériel, et plus précisément du type de processeur. Chaque noyau de processeur peut contenir une seule thread d'impression. Cette valeur par défaut est modifiable. Pour plus d'informations, consulter l'article Traitement parallèle dans le guide utilisateur de NiceLabel Automation.
- **Mode asynchrone.** Dès que le pré-traitement du déclencheur se termine et que les instructions pour le moteur d'impression sont disponibles, la thread d'impression démarre en arrière-plan. Le contrôle est rendu au déclencheur pour qu'il puisse accepter le flux de données entrantes suivant dès que possible. Quand le mode synchrone est activé, le contrôle n'est pas rendu au déclencheur tant que le processus d'impression n'est pas terminé. Cela peut prendre un certain temps, mais le déclencheur a l'avantage d'envoyer un retour d'informations à l'application fournissant les données. Pour plus d'informations, consulter l'article Mode synchrone dans le guide utilisateur de NiceLabel Automation.

> **NOTE**
> L'utilisation de l'option **Enregistrer l'erreur dans une variable**, dans **Exécution d'action et Traitement d'Erreurs** ne va pas produire de résultats en mode asynchrone, car le déclencheur ne recevra pas de retour du processus d'impression. Pour récupérer des informations du processus d'impression, il faut activer le mode synchrone.

> **NOTE**
> Si l'action Imprimer l'étiquette est indentée sous l'action Boucler, Automation l'exécute dans un mode de session d'impression. C'est un mode d'optimisation de l'impression qui imprime toutes les étiquettes d'une boucle dans un seul fichier d'impression. Pour plus d'informations, consulter l'article Impression en session dans le guide utilisateur de NiceLabel Automation.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Quantité : Ce groupe définit le nombre d'étiquettes à imprimer avec le formulaire actif.

- **Étiquettes :** Détermine le nombre d'étiquettes imprimées. **Source de données** spécifie ou ajoute une variable qui définit dynamiquement la quantité d'étiquettes à imprimer.

> **NOTE**
> La valeur de variable est généralement assignée par l'action **Utiliser un Filtre de Données** et doit être un nombre entier.

Toutes (quantité illimitée) : Imprime une quantité d'étiquettes en fonction du masque créé.

Détails d'impression Quantité illimitée

Cette option est utilisée dans deux scénarios.

1. Commander à l'imprimante d'imprimer la même étiquette continuellement jusqu'à ce qu'elle soit éteinte, ou qu'elle reçoive l'ordre d'effacer sa mémoire tampon.

   > **AVERTISSEMENT**
   > Ce scénario requiert l'installation et l'utilisation du pilote d'imprimante NiceLabel pour l'impression d'étiquettes.

   Lors de l'impression d'une étiquette fixe, une seule tâche d'impression est envoyée à l'imprimante, avec la quantité définie sur « illimitée ». Les imprimantes d'étiquettes ont un paramètre de commande d'impression pour indiquer l'impression « illimitée ».
   Quand l'étiquette n'est pas fixe mais qu'elle comporte des objets qui changent durant l'impression, comme des compteurs, alors la quantité imprimée sera définie par la quantité maximale supportée par l'imprimante. Le pilote d'imprimante NiceLabel connaît la quantité limite et imprime toutes les étiquettes possibles.

   > **Exemple**
   > La quantité maximale supportée par l'imprimante est de 32 000. C'est la quantité d'étiquettes qui est imprimée quand l'option Toutes (quantité illimitée) est sélectionnée.

2. Le déclencheur ne fournit aucune donnée, mais il signale seulement que « l'événement s'est effectué ». La logique pour collecter les données nécessaires se trouve sur l'étiquette. Généralement, une connexion vers une base de données est configurée avec l'étiquette et, à chaque déclenchement, l'étiquette doit se connecter à la base de données et récupérer tous les enregistrements. Dans ce cas, l'option **Toutes (quantité illimitée)** se comprend comme « imprimer tous les enregistrements de la base de données ».

- **Quantité variable (définie à partir d'une variable d'étiquette) :** Spécifie une variable de l'étiquette qui va définir la quantité d'étiquettes à imprimer.
  Le déclencheur ne reçoit pas le nombre d'étiquettes à imprimer, donc il transmet la décision au masque de l'étiquette. L'étiquette peut contenir une connexion à la base de données, qui fournira la quantité d'étiquettes, ou il y a une autre source d'information de quantité. Une variable d'étiquette doit être définie comme « variable de quantité ».

Le groupe **Avancé** définit les détails de l'impression d'étiquettes. Cliquer sur **Afficher les options avancées de l'imprimante** pour définir les options d'impression **Avancées** :

Cette section spécifie les paramètres de quantité liés aux quantités d'étiquettes moins fréquemment utilisés.

- **Nombre d'étiquettes sautées :** Définit le nombre d'étiquettes à sauter sur la première page des étiquettes. La feuille d'étiquettes peut avoir déjà été imprimée, mais pas entièrement. Cette feuille peut être utilisée en déplaçant la position de départ. Cette option est applicable pour imprimer les étiquettes sur des feuilles, pas sur des rouleaux d'étiquettes, c'est donc applicable aux imprimantes de bureau mais pas aux imprimantes d'étiquettes.
- **Copies d'étiquettes identiques :** Spécifie le nombre de copies d'étiquettes à imprimer pour chaque étiquette unique. Pour des étiquettes fixes, cette option donne le même résultat que l'option principale **Nombre d'étiquettes**. Avec des étiquettes variables, telles que les étiquettes utilisant des compteurs, ce sera le nombre réel de copies.
- **Nombre de jeux d'étiquettes** spécifie combien de fois la totalité de l'impression d'étiquettes doit se répéter.

Exemple

Le déclencheur, ou l'événement, reçoit un contenu de 3 lignes de données formatées en CSV, donc on attend l'impression de 3 étiquettes (1, 2, 3). En mettant cette option à 3, l'impression se fera dans l'ordre suivant : 1, 2, 3, 1, 2, 3, 1, 2, 3.

- **Métadonnées** : avec chaque travail d'impression, écrit vos commentaires d'impression au Control Center. Vos métadonnées figurant dans **Historique** > **Impression** > colonne **Imprimer les métadonnées**. Vous pouvez utiliser les métadonnées pour le triage et le filtrage, ainsi que pour d'autres fonctions dans le Control Center. Les métadonnées n'ont aucune incidence sur vos impressions ni sur les flux d'impression. Vous utilisez les métadonnées pour enregistrer vos informations supplémentaires concernant vos travaux d'impression dans le Control Center. Les métadonnées peuvent comprendre le numéro de LOT ou d'autres variables d'étiquette, les noms d'imprimantes et les valeurs générées par l'utilisateur/le système.

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

ASTUCE

Toutes les valeurs du groupe Avancé peuvent être soit codées en dur, soit fournies dynamiquement par une variable nouvelle ou existante.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (**`vrai`** ou **`faux`**). Quand le résultat de l'expression est **`vrai`**, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes **`ActionLastErrorId`** et **`ActionLastErrorDesc`**.

5.10.3. Exécuter le fichier de commande Oracle XML

> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent d'utiliser **LMS Enterprise**.

Exécute l'impression avec les données d'un fichier au format Oracle XML.

NiceLabel Automation est directement compatible avec les fichiers XML ayant la structure Oracle XML, qui sont définis par le logiciel Oracle Warehouse Management.

Cette action est un raccourci. Elle permet d'exécuter directement des fichiers Oracle XML sans avoir besoin de les analyser avec un filtre et de mapper les valeurs aux variables.

Pour pouvoir utiliser cette action, le fichier XML doit être conforme au spécifications Oracle XML. Pour plus d'informations, consulter la section Spécifications d'Oracle XML dans le guide utilisateur de NiceLabel Automation.

Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter la section Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

INFO NIVEAU DE PRODUIT

Nécessite **Automation Builder**.

Comment recevoir un fichier de commande dans un déclencheur et l'exécuter

Quand le déclencheur reçoit le fichier de commande et qu'il faut l'exécuter, effectuer les opérations suivantes :

1. Dans le module Automation Builder, dans l'onglet **Variables** , cliquer sur le bouton **Variable Interne**sur le ruban.
2. Dans le menu déroulant, activer la variable interne **`DataFileName`**. Cette variable interne fournit le chemin et le nom du fichier qui contient les données reçues par le déclencheur. Dans ce cas, le contenu est le fichier de commande. Pour plus d'informations, consulter l'article Variables internes dans le guide utilisateur de NiceLabel Automation.
3. Dans l'onglet **Actions**, ajouter l'action pour exécuter le fichier de commande, telle qu'Exécuter un Fichier de Commande, Exécuter un Fichier de Commande Oracle XML, ou Exécuter un Fichier de Commande SAP AII XML.
   Pour l'action **Lancer le Fichier de Commande**, sélectionner le type de fichier de commande dans **Type de Fichier**.
4. Activer l'option **Variable**.
5. Sélectionner la variable **`DataFileName`** dans la liste de variables disponibles.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Fichier** définit le fichier de commande Oracle XML à utiliser.

- **Nom du fichier** : Sélectionne le fichier de commande Oracle XML. Il peut être soit codé en dur, soit fourni dynamiquement par une variable nouvelle ou existante.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.10.4. Exécuter le fichier de commande SAP AII XML

> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action exécute l'impression avec les données d'un fichier au format SAP AII XML.

NiceLabel Automation est directement compatible avec les fichiers XML ayant la structure "SAP AII XML", qui sont définis par le logiciel SAP.

Cette action est un raccourci. Elle permet d'exécuter directement des fichiers SAP AII XML sans avoir besoin de les analyser avec un filtre. Pour pouvoir utiliser cette action, le fichier XML doit être conforme aux spécifications SAP AII XML. Pour plus d'informations, consulter le chapitre Spécifications SAP AII XML dans le guide utilisateur de NiceLabel Automation.

Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter la section Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

INFO NIVEAU DE PRODUIT

Nécessite **Automation Builder**.

Comment recevoir un fichier de commande dans un déclencheur et l'exécuter

Quand le déclencheur reçoit le fichier de commande et qu'il faut l'exécuter, effectuer les opérations suivantes :

1. Dans le module Automation Builder, dans l'onglet **Variables** , cliquer sur le bouton **Variable Interne**sur le ruban.
2. Dans le menu déroulant, activer la variable interne **`DataFileName`**. Cette variable interne fournit le chemin et le nom du fichier qui contient les données reçues par le déclencheur. Dans ce cas, le contenu est le fichier de commande. Pour plus d'informations, consulter l'article Variables internes dans le guide utilisateur de NiceLabel Automation.
3. Dans l'onglet **Actions**, ajouter l'action pour exécuter le fichier de commande, telle qu'Exécuter un Fichier de Commande, Exécuter un Fichier de Commande Oracle XML, ou Exécuter un Fichier de Commande SAP AII XML.
   Pour l'action **Lancer le Fichier de Commande**, sélectionner le type de fichier de commande dans **Type de Fichier**.
4. Activer l'option **Variable**.
5. Sélectionner la variable **`DataFileName`** dans la liste de variables disponibles.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Fichier** définit le fichier de commande SAP AII XML à utiliser.

- **Nom du fichier** : sélectionne le fichier de commande SAP AII XML. Il peut être soit codé en dur, soit fourni dynamiquement par une variable nouvelle ou existante.

Le groupe**Paramètres facultatifs** permet de définir le nom de l'étiquettes s'il n'est pas inclus dans le fichier XML.

- **Nom de l'étiquette** : Le fichier de l'étiquette sélectionnée pour être utilisée dans le fichier de commande. Il peut être soit codé en dur, soit fourni dynamiquement par une variable nouvelle ou existante.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.10.5. Exécuter le fichier de commande

Cette action exécute les commandes du fichier de commande sélectionné. Toutes les options de **Type de fichier** comportent les commandes qu'NiceLabel 10 exécutera de haut en bas.

Les fichiers de commande fournissent généralement des données pour une seule étiquette. Il est possible de définir des fichiers plus complexes. Pour plus d'informations, consulter le chapitre Types de fichiers de commande.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Fichier** définit le type et le nom du fichier de commande à exécuter (JOB, XML or CSV).

- **Type de fichier.** Spécifie le type de fichier de commande à exécuter.
- **Nom du fichier.** Spécifie le nom du fichier de commande.
  Le **Nom du fichier** peut être codé en dur, et le même fichier de commande sera utilisé à chaque fois. L'option **Variable** active le nom de fichier variable. Sélectionner ou créer une variable qui contient le chemin et/ou le nom du fichier, si un déclencheur s'exécute ou un événement survient. En général, la valeur est assignée à la variable par un filtre.
  Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter la section Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

INFO NIVEAU DE PRODUIT

Nécessite **Automation Builder**.

Comment recevoir un fichier de commande dans un déclencheur et l'exécuter

Quand le déclencheur reçoit le fichier de commande et qu'il faut l'exécuter, effectuer les opérations suivantes :

1. Dans le module Automation Builder, dans l'onglet **Variables** , cliquer sur le bouton **Variable Interne**sur le ruban.
2. Dans le menu déroulant, activer la variable interne **`DataFileName`**. Cette variable interne fournit le chemin et le nom du fichier qui contient les données reçues par le déclencheur. Dans ce cas, le contenu est le fichier de commande. Pour plus d'informations, consulter l'article Variables internes dans le guide utilisateur de NiceLabel Automation.

3. Dans l'onglet **Actions**, ajouter l'action pour exécuter le fichier de commande, telle qu'Exécuter un Fichier de Commande, Exécuter un Fichier de Commande Oracle XML, ou Exécuter un Fichier de Commande SAP AII XML.
   Pour l'action **Lancer le Fichier de Commande**, sélectionner le type de fichier de commande dans **Type de Fichier**.
4. Activer l'option **Variable**.
5. Sélectionner la variable **`DataFileName`** dans la liste de variables disponibles.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (**`vrai`** ou **`faux`**). Quand le résultat de l'expression est **`vrai`**, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

  NOTE
  Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes **`ActionLastErrorId`** et **`ActionLastErrorDesc`**.

5.10.6. Envoyer des commandes personnalisées

Cette action exécute les commandes personnalisées saisies dans NiceLabel.

Cette action doit toujours être indentée dans l'action Ouvrir l'étiquette. Elle comporte toutes les références de l'étiquette à laquelle les commandes s'appliquent. Pour plus d'informations, consulter l'article Utilisation des commandes personnalisées dans le guide utilisateur de NiceLabel Automation.

> **NOTE**
> La majorité des commandes personnalisées sont disponibles avec des actions individuelles, donc dans la plupart des cas, il n'y a pas besoin de commandes personnalisées.

> **NOTE**
> L'action Envoyer des commandes personnalisées est utilisable pour terminer une impression en session. C'est un mode d'optimisation de l'impression qui imprime toutes les étiquettes d'une boucle dans un seul fichier d'impression. Pour terminer une impression en session, indenter l'action Envoyer une commande personnalisée sous l'action Boucler et utiliser la commande SESSIONEND. Pour plus d'informations, consulter les articles Impression en session et Utiliser des commandes personnalisées dans le guide utilisateur de NiceLabel Automation.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

L'éditeur de **Script** propose les fonctionnalités suivantes :

- **Insérer une source de données** : Permet d'insérer une variable, nouvelle ou existante, dans un script.
- **Éditeur de script** : Ouvre l'éditeur qui rend l'écriture du script plus facile et plus efficace.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.11. Imprimante

5.11.1. Définir l'imprimante

Cette action spécifie le nom de l'imprimante à utiliser pour imprimer l'étiquette active.

> **NOTE**
> Cette action écrase l'imprimante définie dans les propriétés de l'étiquette.

Cette action est utile pour imprimer des étiquettes identiques sur plusieurs imprimantes. Il faut toujours indenter cette action sous l'action Ouvrir l'étiquette pour donner à l'étiquette la référence de l'imprimante.

Cette action lit les paramètres par défaut (tels que la vitesse et le contraste) du pilote d'imprimante sélectionné et les applique à l'étiquette. À défaut d'utiliser l'action **Installer l'imprimante**, l'étiquette s'imprime comme définie dans le masque d'étiquette.

> **AVERTISSEMENT**
> Attention en passant d'une marque d'imprimante à une autre, par ex. de Zebra à SATO, ou même d'un modèle d'imprimante à un autre modèle de la même marque. Les paramètres d'imprimante peuvent ne pas être compatibles et l'impression de l'étiquette peut ne pas être identique. De même, les optimisations de l'étiquette pour l'imprimante originale, tels que les compteurs internes et les types de caractères internes, peuvent ne pas être disponibles pour l'imprimante sélectionnée.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Imprimante** spécifie le nom de l'imprimante à utiliser pour l'impression en cours.

- **Nom de l'imprimante** : Sélectionner l'imprimante dans la liste d'imprimantes installées localement, ou saisir un nom d'imprimante. Sélectionner **Source de données** pour sélectionner dynamiquement l'imprimante en utilisant une variable. Dans ce cas, sélectionner ou créer une variable qui contient le nom de l'imprimante à utiliser quand l'action s'exécute.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.11.2. Définir le nom du travail d'impression

Cette action spécifie le nom du travail d'impression tel qu'il apparaît dans le spouleur Windows. Par défaut c'est le nom du fichier d'étiquette utilisé. Cette action l'écrasera.

> **NOTE**
> Il faut toujours indenter cette action sous l'action Ouvrir l'étiquette pour qu'elle s'applique à un fichier spécifique.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Travail d'impression** permet de définir le nom du travail d'impression.

- **Nom** : spécifie le nom du travail d'impression. Il peut être codé-en-dur, et le même nom sera utilisé pour chaque action d'Impression. L'option Variable active le nom d'un fichier variable. Sélectionner ou créer une variable qui contient le chemin et/ou le nom du fichier, si un déclencheur s'exécute ou un événement survient.

> **NOTE**
> Dans le module Automation Builder, en général, la valeur de la variable est assignée par un filtre.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.11.3. Rediriger l'impression vers un fichier

Cette action envoie le travail d'impression dans un fichier. Au lieu d'envoyer le fichier d'impression créé au port d'imprimante, comme défini dans le pilote d'imprimante, l'impression est redirigée vers un fichier. Il est possible de joindre des données au fichier existant ou de l'écraser.

Cette action permet de capturer les commandes d'impression dans un fichier distinct.

Cette action demande au module Automation Builder de rediriger l'impression, donc les étiquettes ne s'imprimeront pas. Vérifier qu'elle est suivie par l'action Imprimer l'étiquette.

> **NOTE**
> NiceLabel Automation fonctionne comme un service sous un compte utilisateur Windows défini. Il faut que ce compte utilisateur dispose des privilèges d'accès au fichier spécifié avec les droits de lecture/d'écriture. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur deNiceLabel Automation.

> **NOTE**
> Cette action Rediriger l'impression vers un fichier est également utile pour imprimer plusieurs étiquettes différentes (fichiers .NLBL) sur l'imprimante réseau en conservant l'ordre d'étiquettes correct. Quand plusieurs fichiers .NLBL sont imprimés par le même déclencheur, Automation Builder envoie chaque étiquette dans un travail d'impression distinct à l'imprimante, même si l'imprimante est identique pour les deux étiquettes. Avec une imprimante utilisée en réseau, des travaux provenant d'autres utilisateurs peuvent s'insérer entre deux travaux que le déclencheur doit envoyer ensemble. L'utilisation de cette action permet de combiner les données d'impression dans un même fichier pour envoyer ensuite tout son contenu à l'imprimante en utilisant l'action Envoyer les données à l'imprimante.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe de paramètres de **Fichier** définit comment se fait la sélection du fichier pour la redirection.

- **Nom du fichier** : Spécifie le nom du fichier. Il peut être soit codé en dur, soit fourni dynamiquement par une variable nouvelle ou existante.
  Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

> **NOTE**
> Il faut que ce compte utilisateur dispose des privilèges d'accès au fichier spécifié avec les droits de lecture/d'écriture.

Le groupe de paramètres **Mode d'écriture du fichier** permet de sélectionner comment le fichier sera traité en cas de redirections répétées.

- **Écraser le fichier** : Si le fichier spécifié existe déjà sur le disque, il sera écrasé.
- **Joindre les données au fichier** : Les données du travail d'impression sont ajoutées aux données existantes dans le fichier donné.

Le groupe **Persistance** permet de contrôler la continuité de l'action de redirection. Il permet de définir le nombre d'actions Imprimer l'étiquette qui sont concernées par l'action **Rediriger l'impression vers un fichier**.

- **Appliquer à la prochaine action d'impression** : Précise que la redirection de l'impression ne s'appliquera qu'à la prochaine action Imprimer l'étiquette (un seul événement).
- **Appliquer à toutes les actions d'impression suivantes** : Précise que la redirection de l'impression s'appliquera à toutes les actions **Imprimer l'étiquette** définies après la présente action **Rediriger l'impression vers un fichier**.

> **NOTE**
> L'action redirige seulement l'impression. Vérifier qu'elle est suivie par l'action Imprimer l'étiquette.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

NOTE

Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.11.4. Définir les paramètres d'impression

INFO NIVEAU DE PRODUIT

Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action permet d'affiner les paramètres relatifs au pilote d'imprimante. Ces paramètres concernent la vitesse et le contraste des imprimantes d'étiquettes, ou le réservoir à papier pour les imprimantes laser.

Les paramètres d'imprimante s'appliquent uniquement à l'impression en cours. Ils ne sont pas mémorisés pour les déclencheurs suivants.

AVERTISSEMENT

Vos paramètres à partir de l'action **Définir les paramètres d'impression** ne s'appliquent pas lorsque vous prévisualisez ou réimprimez les étiquettes à partir de votre Control Center.

Vous pouvez éviter ceci en enregistrant vos paramètres d'impression (propriétés de l'imprimante) dans votre modèle d'étiquette ou votre pilote d'imprimante. Les propriétés de l'imprimante peuvent varier d'un pilote à l'autre, et également d'une imprimante à l'autre.

NOTE

Après l'action Définir l'imprimante pour changer l'imprimante, il faut ensuite utiliser l'action **Définir les paramètres d'impression**. Avant d'appliquer la structure DEVMODE au pilote d'imprimante, il faut commencer par charger les paramètres de l'imprimante par défaut. Ce qui sera effectué par l'action Installer l'imprimante. Le DEVMODE est compatible uniquement avec le DEVMODE du même pilote d'imprimante.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Paramètres d'impression** permet d'affiner les paramètres avant l'impression.

- **Bac** : Nom du bac contenant le support d'étiquettes. Cette option est utilisée généralement avec les imprimantes laser et à jet d'encre ayant différents bacs à papier. Le nom du bac à papier fourni doit correspondre au nom du bac du pilote d'imprimante. Vérifier les propriétés du pilote d'imprimante pour plus de détails.
- **Vitesse d'impression** : Définit la vitesse d'impression. Ce paramètre écrase celui qui est défini dans l'étiquette. La valeur fournie doit se situer dans la gamme de valeurs acceptables.

Exemple

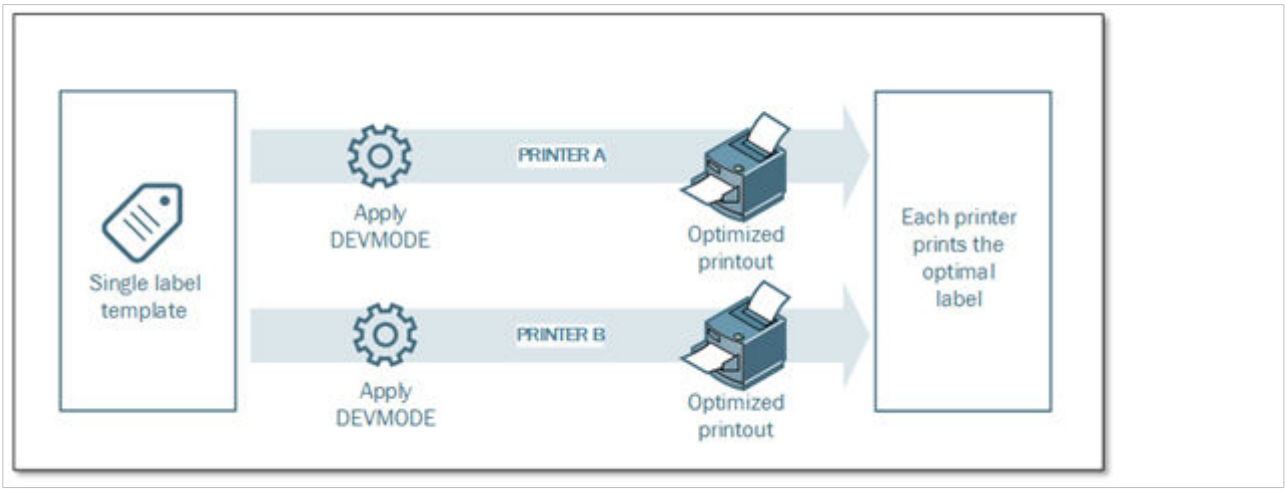
Le premier modèle d'imprimante accepte une gamme de valeurs de 0 à 30, tandis que le second modèle d'imprimante accepte des valeurs de -15 à 15. Pour plus d'informations, voir les propriétés du pilote d'imprimante

- **Contraste** : Définit le contraste des objets imprimés sur le papier et écrase les paramètres de l'étiquette. La valeur fournie doit se situer dans la gamme de valeurs acceptée par l'imprimante.
- **Décalage d'impression X** : Applique le décalage horizontal. L'impression de l'étiquette sera déplacée horizontalement du nombre de pixels spécifiés. Un décalage négatif est possible.
- **Décalage d'impression Y** : Applique le décalage vertical. L'impression de l'étiquette sera déplacée verticalement du nombre de pixels spécifiés. Un décalage négatif est possible.

ASTUCE

Tous les paramètres d'impression peuvent être soit codés en dur, soit fournis dynamiquement par une variable nouvelle ou existante.

Le groupe **Avancé** permet de personnaliser les paramètres d'impression envoyés avec le travail d'impression.



Tous les **paramètres de l'imprimante**, tels que la vitesse d'impression, le contraste, le type de média, le décalage et autres peuvent être définis comme suit :

- Définis dans une étiquette
- Rappelés depuis le pilote d'imprimante
- Rappelés depuis l'imprimante au moment de l'impression

Les méthodes utilisables dépendent du pilote d'imprimante et des possibilités de l'imprimante. Le mode d'impression (Rappel des paramètres de l'étiquette, du pilote ou de l'imprimante) est configurable à la conception de l'étiquette. Il faut parfois appliquer ces paramètres d'imprimante au moment de l'impression. Ils peuvent varier à chaque impression.

Exemple

Une seule étiquette peut être imprimée sur différentes imprimantes, mais chaque imprimante nécessite des paramètres légèrement différents. Les imprimantes de différents fabricants n'utilisent pas les mêmes valeurs de vitesse ou de température. De plus, certaines imprimantes requièrent un décalage vertical ou horizontal pour imprimer l'étiquette au bon endroit. Durant la phase de test, il faut déterminer les meilleurs paramètres pour chaque imprimante utilisée et les appliquer à un masque d'étiquette unique juste avant d'imprimer. Cette action va appliquer les paramètres correspondants pour chaque imprimante définie.

Cette action requiert la réception des paramètres d'impression sous une structure DEVMODE. C'est une structure de données aux normes Windows comportant les informations concernant l'initialisation et l'environnement d'une imprimante.

L'option **Paramètres d'imprimante** appliquera les paramètres d'imprimante personnalisés. Liste des entrées disponibles :

- **DEVMODE encodé en base64 - Données fixes**. Dans ce cas, Il faut fournir le DEVMODE de l'imprimante encodé dans une chaîne de caractères Base64 directement dans le champ d'édition. Quand l'action est exécutée, elle convertit les données encodées en Base64 sous forme binaire.

- **DEVMODE encodé en base64 - Données variables**. Dans ce cas, la source de données sélectionnée doit contenir le DEVMODE encodé Base64. Activer la source de données et sélectionner la variable appropriée dans la liste. Quand l'action est exécutée, elle convertit les données encodées en Base64 sous forme binaire.
- **DEVMODE binaire - données variables (disponible dans Automation Builder)**. Dans ce cas, la variable sélectionnée doit contenir le DEVMODE sous sa forme binaire native. Activer la **source de données** et sélectionner la variable appropriée dans la liste. Quand elle est exécutée, l'action va utiliser le DEVMODE tel quel, sans aucune conversion.

> **NOTE**
> Si la variable fournit un DEVMODE binaire, il faut que la variable sélectionnée soit définie comme variable binaire dans la configuration du déclencheur.

> **NOTE**
> Vérifier que l'action Installer l'imprimante est définie avant cette action.

Paramètres d'étiquette annule les propriétés d'étiquette définies dans **Propriétés de l'étiquette** dans Designer. Utilisez cette option quand vous imprimez vos étiquettes sur une imprimante ou un support avec différentes propriétés, selon la définition figurant dans **Propriétés de l'étiquette** dans Designer. Avec cette option, vous pouvez :

- Modifier les dimensions de vos étiquettes (largeur et hauteur).
- Modifiez les marges des étiquettes ou en ajouter.
- Désactiver le massicot.
- Désactiver l'impression par lot.
- Appliquer différents stocks en modifiant le paramètre **Etiquettes en transverse** (compte horizontal et vertical count, écarts, ordre de traitement).
- Redéfinir l'orientation Portrait ou Paysage.
- Faire pivoter les étiquettes de 180°.

Automation appliquer les **Paramètres de l'étiquette** au moment de l'impression. Les paramètres de l'étiquette ne sont pas enregistrés dans vos modèles d'étiquettes. Vous pouvez fournir des paramètres d'étiquettes sous forme de données utiles XML.

Exemple de paramètres d'étiquette au format XML

Cet exemple présente une vue structurelle des paramètres d'étiquette et de leurs attributs.

NOTE

Les unités dimensionnelles dans XML correspondent aux unités de conception de vos étiquettes (cm, po., mm, points). Vous pouvez modifier les unités dans Designer, en allant dans **Propriétés de l'étiquette** > **Dimensions de l'étiquette** > **Unité de mesure**.

```
<LabelSettings>
    <Width>100</Width>
    <Height>30</Height>
    <Margin>
        <Left>2</Left>
        <Right>3</Right>
        <Top>4</Top>
        <Bottom>5</Bottom>
    </Margin>
    <LabelsAcross>
        <Horizontal>
            <Count>2</Count>
            <Gap>4</Gap>
        </Horizontal>
        <Vertical>
            <Count>3</Count>
            <Gap>5</Gap>
        </Vertical>
        <ProcessingOrder>HorizontalTopRight</ProcessingOrder>
    </LabelsAcross>
    <Orientation>Landscape</Orientation>
    <Rotated>true</Rotated>
    <DisableCutter/>
    <DisableBatchPrinting/>
</LabelSettings>
```

Spécification XML des paramètres d'étiquette

Cette section contient une description de la structure de fichier XML pour définir les réglages et les valeurs des **Paramètres de l'étiquette**.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="nonNegativeFloat">
    <xs:restriction base="xs:float">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="LabelSettings">
    <xs:complexType>
```

```
      <xs:all>
        <xs:element name="DisableCutter" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence/>
          </xs:complexType>
        </xs:element>
        <xs:element name="DisableBatchPrinting" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence/>
          </xs:complexType>
        </xs:element>
        <xs:element name="Width" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Height" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Margin" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element name="Left" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
              <xs:element name="Right"  type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
              <xs:element name="Top"  type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
              <xs:element name="Bottom" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element name="LabelsAcross" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element name="Horizontal" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="Count" type="xs:nonNegativeInteger"
minOccurs="0" maxOccurs="1" />
                    <xs:element name="Gap" type="nonNegativeFloat"
minOccurs="0" maxOccurs="1"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
              <xs:element name="Vertical" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="Count" type="xs:nonNegativeInteger"
```

```
minOccurs="0" maxOccurs="1" />
                    <xs:element name="Gap" type="nonNegativeFloat"
minOccurs="0" maxOccurs="1"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
              <xs:element name="ProcessingOrder" minOccurs="0" maxOccurs="1">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:enumeration value="HorizontalTopLeft"/>
                    <xs:enumeration value="HorizontalTopRight"/>
                    <xs:enumeration value="HorizontalBottomLeft"/>
                    <xs:enumeration value="HorizontalBottomRight"/>
                    <xs:enumeration value="VerticalTopLeft"/>
                    <xs:enumeration value="VerticalTopRight"/>
                    <xs:enumeration value="VerticalBottomLeft"/>
                    <xs:enumeration value="VerticalBottomRight"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element name="Orientation" minOccurs="0" maxOccurs="1">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Portrait"/>
              <xs:enumeration value="Landscape"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Rotated" type="xs:boolean" minOccurs="0"
maxOccurs="1"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.

- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.11.5. Rediriger l'impression vers un PDF

> **INFO NIVEAU DE PRODUIT**
> Cette option est disponible dans NiceLabel LMS Enterprise.

Cette action envoie le travail d'impression dans un fichier PDF. Le document PDF aura les dimensions exactes de l'étiquette, comme définies à la conception de l'étiquette. La qualité du rendu des graphiques dans les PDF correspond à la résolution de l'imprimante ciblée et aux dimensions d'impression désirées.

Les données du flux d'impression peuvent être jointe à un fichier existant ou l'écraser.

Cette action demande au module NiceLabel 10 de rediriger l'impression - donc les étiquettes ne s'imprimeront pas. Vérifier qu'elle est suivie par l'action **Imprimer l'étiquette**.

> **NOTE**
> Le module NiceLabel Automation tourne en service sous un compte utilisateur Windows défini. Il faut que ce compte utilisateur dispose des privilèges d'accès au fichier spécifié avec les droits de lecture/d'écriture. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Fichier** définit le fichier à ouvrir.

- **Nom de fichier** : spécifie le fichier dans lequel les données vont s'imprimer. Il peut être codé en dur, et l'impression sera redirigée vers le même fichier à chaque fois. Pour en définir un dynamiquement, utiliser une variable existante ou en créer une nouvelle.
- **Écraser le fichier** : Si le fichier spécifié existe déjà sur le disque, il sera écrasé (sélectionné par défaut).
- **Joindre les données au fichier** : Les données du travail d'impression sont ajoutées aux données existantes dans le fichier fourni (désélectionne par défaut).
- **Intégrer des polices au format PDF** : Si vous utilisez une police non standard, votre solution peut produire différentes sorties PDF sur des ordinateurs ne prenant pas en charge cette police. L'option **Intégrer des polices au format PDF** étant activée, vous pouvez intégrer votre police non standard à votre solution et cette dernière créé le même PDF sur tous les ordinateurs.
  L'option **Intégrer des polices au format PDF** créé un PDF pour le document conforme à l'archivage (PDF/A). La norme PDF/A veille à ce que vos documents soient reproduits exactement de la même manière, indépendamment du logiciel que vous utilisez. Les informations nécessaires pour l'affichage de contenu, comme des images, des polices et des informations relatives à la couleur sont intégrées dans votre fichier PDF.

  > **NOTE**
  > Le fichier de sortie PDF consomme plus d'espace lorsqu'il est enregistré avec cette option activée.

Le groupe **Persistance** permet de contrôler la continuité de l'action de redirection. Elle permet de définir le nombre d'actions Imprimer l'étiquette qui sont concernées par l'action **Rediriger l'impression vers un fichier**.

- **Appliquer à la prochaine action d'impression** : Précise que la redirection de l'impression ne s'appliquera qu'à la prochaine action Imprimer l'étiquette (un seul événement).
- **Appliquer à toutes les actions d'impression suivantes** : Précise que la redirection de l'impression s'appliquera à toutes les actions Imprimer l'étiquette définies après la présente action Rediriger l'impression vers un fichier.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.11.6. État de l'imprimante

> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action communique avec l'imprimante pour récupérer en temps réel son état, et contacte le spouleur de Windows pour récupérer d'autres informations sur l'imprimante et les travaux en cours.

Il en résulte la collecte d'informations concernant les erreurs, l'état de la file d'attente, le nombre de travaux dans la file d'attente. ce qui révèle les erreurs potentielles et permet de les identifier facilement.

> **NOTE**
> Scénarios possibles. (1) Vérification de l'état de l'imprimante avant l'impression. Si l'imprimante est dans un état d'erreur, imprimer l'étiquette sur l'imprimante en réserve. (2) Compter le nombre de travaux en attente dans un spouleur de l'imprimante principale. S'il y en a trop, vous imprimerez l'étiquette sur une imprimante alternative. (3) Pour vérifier l'état de l'imprimante avant l'impression. Si l'imprimante est dans un état d'erreur, vous ne pouvez pas imprimer d'étiquettes, mais l'erreur est envoyée au système principal en utilisant une des actions de sortie, telles que Envoyer les données au port TCP/IP, Envoyer les données à HTTP, Exécuter une requête SQL, Service Web, ou comme réponse du déclencheur.

Prérequis pour état de l'imprimante en temps réel

Pour pouvoir contrôler l'état de l'imprimante en temps réel, procéder comme suit :

- Utiliser un pilote d'imprimante NiceLabel pour recevoir les informations détaillées sur le statut de l'imprimante. Avec un pilote d'imprimante différent, seules les informations provenant du spouleur Windows sont contrôlables.
- L'imprimante doit être capable de rapporter son état en temps réel. Pour les modèles d'imprimantes supportant la communication bidirectionnelle, voir la page Internet Téléchargement NiceLabel.
- L'imprimante doit être connectée à une interface bidirectionnelle.
- Le mode bidirectionnel doit être activé dans le **Panneau de configuration > Matériel et audio > Périphériques et Imprimantes > pilote > Propriétés de l'imprimante > Onglet Ports > Activer la gestion du mode bidirectionnel**.
- Avec une imprimante connectée au réseau, vérifier que le port utilisé est le **Port TCP/IP Avancé** et pas le **Port TCP/IP Standard**. Pour plus d'informations, consulter la Base de Connaissances article.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.

- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Imprimante** sélectionne l'imprimante.

- **Nom de l'imprimante** spécifie le nom de l'imprimante à utiliser pour l'impression en cours. Sélectionner l'imprimante dans la liste d'imprimantes installées localement, ou saisir le nom d'une imprimante. Une source de données active un nom d'imprimante variable. Dans ce cas, sélectionner ou créer une variable qui contient le nom de l'imprimante quand un déclencheur s'exécute ou qu'un événement survient. En général, la valeur de la variable est assignée par un filtre.

Le groupe **Mappage de données** détermine les paramètres qui sont renvoyés par l'action **État de l'imprimante**.

> **AVERTISSEMENT**
> La majorité des paramètres suivants n’est compatible qu’avec les pilotes d'imprimante de NiceLabel. Avec un autre pilote d'imprimante, seuls les paramètres relatifs au spouleur sont utilisables.

- **État de l'imprimante :** Spécifie l'état de l'imprimante en temps réel comme une chaîne de caractères. Si l'imprimante rapporte plusieurs états, ils sont fusionnés ensemble dans une chaîne de caractères, délimités par des virgules ",". Si aucun problème d'imprimante n'est reporté, le champ est vide. L'état de l'imprimante peut être : **Hors ligne**, **Plus d'étiquettes** ou **Ruban quasi fini**. Les phrases de rapport ne sont pas normalisées, donc chaque marque d'imprimante peut utiliser des messages d'état différents.
- **Erreur d'impression :** Spécifie la valeur booléenne (vrai/faux) de l'état d'erreur de l'imprimante.
- **Imprimante hors ligne :** Spécifie la valeur booléenne (vrai/faux) de l'état de l'imprimante hors ligne.
- **Pilote en pause :** Spécifie la valeur booléenne (vrai/faux) de l'état de pause du pilote.
- **Pilote NiceLabel :** spécifie la valeur booléenne (vrai/faux) de l'état du pilote de l'imprimante. Dit si le pilote sélectionné est un pilote d'imprimante NiceLabel.
- **État du spouleur :** Spécifie l'état du spouleur sous forme de chaîne rapportée par Windows. Le spouleur peut être simultanément dans différents états. Dans ce cas, les états sont fusionnés avec des virgules ",".
- **ID de l'état du spouleur :** Spécifie l'état du spouleur sous forme de numéro rapporté par Windows. Le spouleur peut être simultanément dans différents états. Dans ce cas, les ID d'état contiennent toutes les ID comme indicateurs. Par exemple, la valeur 5 représente les ID des états 4 et 1, ce qui se traduit par "Imprimante en erreur, Imprimante en pause". Se référer au tableau ci-dessous.

> **ASTUCE**
> L'action renvoie une valeur décimale, les valeurs dans le tableau ci-dessous sont en hexadécimales, il faut donc les convertir avant d'analyser la réponse.

- **Table d'ID d'états du spouleur et descriptions correspondantes**

ID de l'état du spouleur (en hex)	Description de l'état du spouleur
0	Pas d'état.
1	Imprimante en pause.
2	Imprimante en impression.
4	Imprimante en erreur.
8	Imprimante pas disponible.
10	L'imprimante n'a plus de papier.
20	Alimentation manuelle requise.
40	L'imprimante a un problème de papier.
80	Imprimante hors-ligne.
100	État Entrée/Sortie actif.
200	Imprimante occupée.
400	Blocage de papier.
800	Corbeille de sortie pleine.
2000	Imprimante en attente.
4000	Imprimante en cours d'exécution.
10000	Imprimante en pré-chauffage.
20000	Niveau d'encre bas.
40000	Plus d'encre dans l'imprimante.
80000	La page actuelle ne peut pas être imprimée.
100000	Une intervention de l'utilisateur est requise.
200000	L'imprimante n'a plus de mémoire disponible.
400000	Protection ouverte.
800000	Erreur inconnue.
1000000	L'imprimante est en mode de sauvegarde d'énergie.

- **Nombre de travaux dans le spouleur :** Spécifie le nombre de travaux qui sont dans le spouleur pour l'imprimante sélectionnée.
- **État de l'imprimante étendue de pilote NiceLabel :** renvoie l'état de l'imprimante et les données d'erreurs au format JSON. Les données comprennent également les paramètres d'imprimante tels que la version du micrologiciel, la distance de la tête d'impression et le niveau de l'encre.

> **NOTE**
> Il vous faut le dernier pilote NiceLabel pour obtenir l'**état de l'imprimante étendu du pilote NiceLabel**.

Exemple

NiceLabel Automation renvoie le contenu JSON suivant :


```
{
    "printerName":"Production_printer_09",
    "version":"1",
    "responseType":"status3",
    "overallStatus":"error",
    "operationState":"pause",
    "printerType":"Thermal Transfer Label Printer",
    "deviceStatus":
    [
        {
            "item": "fatalError",
            "type": "string",
            "internalId":"ERR_PAPER_OUT",
            "shortString":"Paper Out",
            "longString":"Printer is out of paper."
        },
        {
            "item": "error",
            "type": "string",
            "internalId": "ERR_RIBBON_OUT",
            "shortString": "Ribbon Out",
            "longString": "Printer is out of Ribbon."
        }
    ],
    "deviceParameters":
    [
        {
            "item":"state",
            "type":"float",
            "internalId":"INFO_PRINTHEAD_DISTANCE",
            "unit": "meter",
            "value": 789,
            "shortDescription":"Printhead distance",
            "longDescription":"Distance of printhead usage."
        },
        {
            "item": "state",
            "type": "string",
            "internalId": "INFO_FW",
            "unit": "",
            "value": "FW27.3.13",
            "shortDescription": "Firmware version",
```

```
            "longDescription": "The version of the Firmware loaded on the
printer."
        },
        {
            "item": "property",
            "type": "integer",
            "internalId": "INFO_PRINT_DISTANCE",
            "unit": "seconds",
            "value": 16004,
            "shortDescription": "Print Distance",
            "longDescription": "Print Distance that the printer printed
since odometer reset."
        }
    ]
}
```

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions.

Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.11.7. Stocker l'étiquette sur l'imprimante

INFO NIVEAU DE PRODUIT

Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action enregistre un masque d'étiquette dans la mémoire de l'imprimante. L'action est une partie vitale du mode d'impression Stocker/Rappeler, où le masque d'étiquette est d'abord enregistré dans la mémoire de l'imprimante et ensuite rappelé depuis la mémoire. Les parties non modifiables du masque d'étiquette sont déjà enregistrées dans l'imprimante, il suffit de fournir les données des objets variables de l'étiquette avant l'impression. Pour plus d'informations, consulter l'article Utiliser le mode d'impression Stocker/Rappeler dans le guide utilisateur de NiceLabel Automation.

NOTE

Le temps requis pour le transfert des données à l'imprimante est considérablement raccourci, comme il y a moins de données à envoyer. Cette action est utilisée dans les scénarios d'impression autonomes : l'étiquette est stockée dans l'imprimante ou l'applicateur sur une ligne de production, puis rappelée plus tard par un déclencheur logiciel ou mécanique, tel qu'un lecteur de codes à barres ou une cellule photoélectrique.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Options avancées pour stocker l'étiquette dans l'imprimante** définit le nom de l'imprimante et les variantes de stockage.

- **Nom de l'étiquette à utiliser sur l'imprimante** : Spécifie le nom à utiliser pour la stockage du masque d'étiquette dans la mémoire de l'imprimante. Entrer le nom manuellement ou activer **Source**

de données pour définir le nom de façon dynamique en utilisant une valeur variable existante ou nouvelle.

> **AVERTISSEMENT**
> Lors de l'enregistrement d'une étiquette dans l'imprimante, il est recommandé de laisser vide le nom d'étiquette dans les données avancées. Ceci évite des conflits de noms durant le processus de rappel de l'étiquette.

- **Variante de stockage** : Définit l'emplacement des masques d'étiquettes dans la mémoire de l'imprimante. Entrer l'emplacement manuellement ou activer **Source de données** pour le définir de façon dynamique en utilisant une valeur variable existante ou nouvelle.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.11.8. Imprimer un document PDF

INFO NIVEAU DE PRODUIT

Les fonctions Automation Builder nécessitent d'utiliser **LMS Enterprise**.

L'action Imprimer un document PDF imprime des documents PDF statiques qui ne sont pas liés aux étiquettes dans vos solutions PowerForms ou configurations NiceLabel Automation. Utiliser cette action pour imprimer n'importe quel document PDF directement à partir de vos solutions ou configurations. Les documents PDF peuvent être stockés dans/sur:

- votre ordinateur
- NiceLabel Control Center
- le serveur Web
- les pilotes partagés en réseau

ASTUCE

Cette action est utile si vous prévoyez d'équiper vos colis avec des rapports PDF imprimés sur des éléments inclus, ou si vous voulez imprimer la documentation des emballages sans ouvrir l'explorateur de fichiers.

NOTE

L'action Imprimer un document PDF utilise une imprimante sur votre quota de licences. Lire le document de licence NiceLabel pour plus d'informations sur la gestion des licences.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Imprimante** spécifie le nom de l'imprimante à utiliser pour l'impression en cours.

- **Nom de l'imprimante** : Sélectionner l'imprimante dans la liste d'imprimantes installées localement, ou saisir un nom d'imprimante. Sélectionner **Source de données** pour sélectionner dynamiquement l'imprimante en utilisant une variable. Dans ce cas, sélectionner ou créer une variable qui contient le nom de l'imprimante à utiliser quand l'action s'exécute.

Le groupe **Fichier** définit le fichier à ouvrir.

- **Nom du fichier** : spécifie le PDF à imprimer.

> **NOTE**
> Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.12. Variables

5.12.1. Définir une variable

Cette action assigne une nouvelle valeur à la variable sélectionnée.

Généralement, les variable récupèrent leur valeur par l'action Utiliser du filtre de données (dans Automation Builder) qui extrait des champs dans les données reçues et les relie aux variables. Il est parfois nécessaire de définir personnellement les valeurs de variables, souvent en cas de dépannage. Dans Automation Builder, les valeurs de variables ne sont pas mémorisées d'un déclencheur à l'autre, mais sont conservées durant l'utilisation du déclencheur.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Variables** définit le nom de la variable et sa valeur.

- **Nom** : nom de la variable dont la valeur doit changer.
- **Valeur :** Assigne une nouvelle valeur à la variable sélectionnée.. Elle peut être soit codée en dur, soit fournie dynamiquement par une variable nouvelle ou existante.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes **`ActionLastErrorId`** et **`ActionLastErrorDesc`**.

5.12.2. Enregistrer les données variables

> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action enregistre les valeurs d'une ou plusieurs variables dans le fichier associé.

Dans le module NiceLabel Automation cette action permet d'échanger des données entre les déclencheurs. Pour relire les données dans le déclencheur, utiliser l'action Charger les Données Variables.

> **ASTUCE**
> Les valeurs sont enregistrées au format CSV, avec le nom des variables sur la première ligne. Si les variables contiennent des valeurs mufti lignes, les caractères de la nouvelle ligne (CR/LF) sont encodés ainsi : **`\n\r`**.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.

- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Paramètres** définit le nom du fichier.

- **Nom de fichier** : nom du fichier dans lequel il faut enregistrer les données variables. Il peut être codé en dur, et les valeurs seront enregistrées chaque fois dans le même fichier.
  Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter la section Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

Le groupe **Si le fichier existe** gère les options en cas de fichier existant.

- **Ecraser le fichier** : écrase les données existantes avec les nouvelles données variables. L'ancien contenu est perdu.
- **Joindre les données au fichier** : joint les valeurs de la variable au fichier de données existantes.

Le groupe **Structure du fichier texte** spécifie les paramètres du fichier CSV :

- **Séparateur** : spécifie le type de séparateur (tabulation, point virgule, virgule ou caractère personnalisé). Le séparateur est un caractère qui sépare les valeurs.
- **Délimiteur de texte** : spécifie le caractère qui délimite le contenu du texte.
- **Encodage du fichier** : Spécifie le mode d'encodage utilisé dans le fichier de données. **Auto** définit automatiquement l'encodage. Si nécessaire, sélectionner le type d'encodage préféré dans le menu déroulant.

  > **ASTUCE**
  > UTF-8 est une bonne sélection par défaut.

- **Ajouter les noms des variables dans la première ligne** : place le nom de la variable dans la première ligne du fichier.

Le groupe **Variables** définit les variables dont les valeurs doivent être lues dans le fichier de données. Les valeurs des variables existantes seront remplacées par les valeurs du fichier.

- **Toutes les variables** : spécifie que toutes les variables définies dans le fichier de données devront être lues.
- **Variables sélectionnées** : Spécifie que seules les variables sélectionnées seront lues dans le fichier de données.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.12.3. Charger les données variables

> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action charge les valeurs d'une ou plusieurs variables enregistrées dans le fichier associé par l'action **Enregistrer les données variables** . Cette action permet d'échanger des données entre les déclencheurs. Il est possible de charger une variable particulière ou toutes les variables qui existent dans le fichier.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Paramètres** définit le nom du fichier.

- **Nom de fichier** : Spécifie le fichier duquel il faut charger les données variables. Il peut être codé en dur, et les valeurs seront chargées chaque fois du même fichier.
  Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter la section Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

Les paramètres du groupe **Structure du fichier** doivent refléter la structure du fichier enregistré par l'action Enregistrer les données variables.

- **Séparateur** : Spécifie le type de séparateur (tabulation, point virgule, virgule ou caractère personnalisé). Le séparateur est un caractère qui sépare les valeurs.
- **Délimiteur de texte** : Spécifie le caractère qui délimite le contenu du texte.
- **Encodage du fichier** : Spécifie le mode d'encodage utilisé dans le fichier de données. **Auto** définit automatiquement l'encodage. Si nécessaire, sélectionner le type d'encodage préféré dans le menu déroulant.

> **ASTUCE**
> UTF-8 est une bonne sélection par défaut.

Le groupe **Variables** définit les variables dont les valeurs doivent être chargées.

- **Toutes les variables** : Spécifie que toutes les variables définies dans le fichier de données devront être lues.
- **Variables sélectionnées** : Spécifie que seules les variables sélectionnées seront lues dans le fichier de données.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.

- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

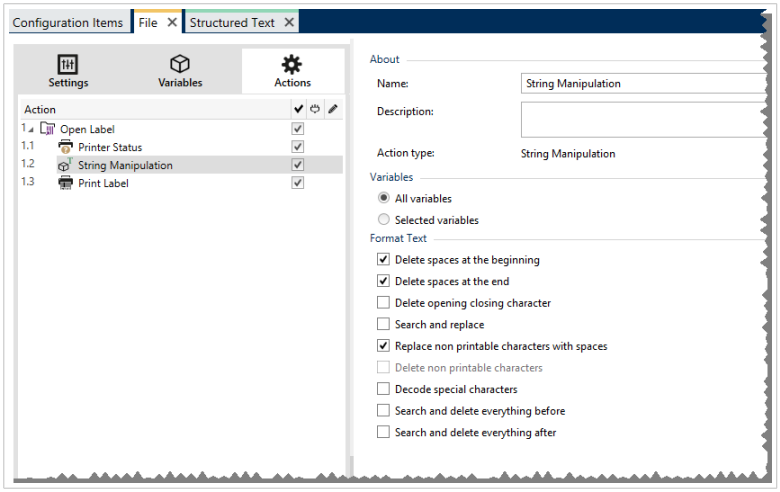
5.12.4. Manipulation de chaîne de caractères

Cette action définit le formatage des données variables sélectionnées.

Les actions les plus utilisées sont : effacer les espaces de début et de fin, rechercher et remplacer des caractères, et effacer les guillemets d'ouverture et de fermeture.

Cette fonctionnalité sert souvent quand un déclencheur reçoit un fichier de données non structuré ou des données anciennes. Dans ce cas, les données doivent être analysées par le filtre de **Données non structurées**. Cette action permet d'affiner la valeur des données.

> **NOTE**
> Parfois, cette action n'est pas assez puissante pour manipuler une chaîne de caractères. Utiliser alors l'action **Exécuter le Script** et un script Visual Basic ou Python pour manipuler les données.



À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Variables** définit les variables dont les valeurs doivent être formatées.

- **Toutes le variables** : Spécifie toutes les variables définies dans le fichier de données qui devront être formatées.
- **Variables sélectionnées** : Spécifie que seules les variables sélectionnées dans le fichier de données seront formatées.

Le groupe **Formater le texte** définit les fonctions de manipulation de chaîne de caractères qui seront appliquées aux variables ou champs sélectionnés. Plusieurs fonctions sont utilisables. Les fonctions s'appliqueront dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début** : Enlève tous les caractères d'espacement (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin** : Enlève tous les caractères d'espacement (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Effacer le caractère d'ouverture fermeture** : Efface la première occurrence du caractère d'ouverture et de fermeture trouvé dans la chaîne de caractères.

Exemple

Si on utilise "{" comme caractère d'ouverture et "}" comme caractère de fermeture, la chaîne d'entrée `{{selection}}` est convertie en `{selection}`.

- **Rechercher et remplacer** : Exécute une recherche classique et remplace la fonction selon la valeur fournie pour rechercher et remplacer par. Les expressions régulières sont utilisables.

> **NOTE**
> Il y a plusieurs implémentations des expressions classiques utilisées. NiceLabel 10 utilise la syntaxe .NET Framework pour les expressions clasiques. Pour plus d'informations, consulter la Base de Connaissances article.

- **Remplacer les caractères non imprimables par des espaces** : remplace tous les caractères de contrôle de la chaîne par un espace (code décimal ASCII 32). Les caractères non imprimables sont des caractères avec des valeurs décimales ASCII entre 0–31 et 127–159.
- **Supprimer les caractères non imprimables** : Enlève tous les caractères de contrôle de la chaîne. Les caractères non imprimables sont des caractères avec des valeurs décimales ASCII entre 0–31 et 127–159.
- **Décoder les caractères spéciaux** : Décode les caractères (ou codes de contrôle) qui sont indisponibles sur le clavier, comme le retour chariot ou le passage à la ligne. NiceLabel 10 utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Chariot et <LF> pour Passage à la Ligne. Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

> **Exemple**
> Quand il reçoit les données "<CR><LF>", Designer les utilise comme une chaîne complète de 8 caractères. Il faut activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires CR (Retour Chariot – code ASCII 13) et LF (Passage à la ligne – code ASCII 10).

- **Rechercher et supprimer tout avant** : Recherche la chaîne et enlève les caractères situés avant la chaîne définie. La chaîne de caractères trouvée peut aussi être effacée.
- **Rechercher et supprimer tout après** : Recherche la chaîne et enlève les caractères situés après la chaîne définie. La chaîne de caractères trouvée peut aussi être effacée.
- **Changement de casse** : Modifie tous les caractères dans vos chaînes en lettres capitales ou minuscules.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.

- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.13. Impression par lot

5.13.1. Boucler

> **INFO NIVEAU DE PRODUIT**
> Cette action est disponible dans **NiceLabel LMS Enterprise.**

Cette action exécute de multiple fois les actions indentées subordonnées. Toutes les actions indentées s'exécutent en boucle autant de fois que défini par la différence entre les valeurs de départ et d'arrivée.

> **NOTE**
> L'action Boucler lance l'impression en session - un mode d'optimisation de l'impression qui imprime toutes les étiquettes d'une boucle dans un seul fichier d'impression. Pour plus d'informations, consulter l'article Impression en session dans le guide utilisateur de NiceLabel Automation.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Paramètres de boucle** comporte les options suivantes :

- **Valeur de départ** : Point de référence pour démarrer la boucle. Sélectionner la Source de données pour définir dynamiquement la valeur de départ en utilisant une valeur variable. Sélectionner une variable contenant une valeur numérique pour le départ.
- **Valeur finale** : point final de référence. Sélectionner la Source de données pour définir dynamiquement la valeur de départ en utilisant une valeur variable. Sélectionner une variable contenant une valeur numérique pour le départ.

  > **ASTUCE**
  > Les **Valeur de départ** et **Valeur finale** peuvent être négatives.

- **Enregistrer la valeur de la boucle dans une variable** : enregistre la valeur du pas de la boucle dans une variable nouvelle ou existante. La valeur du pas de bouclage se situe entre la valeur de départ et la valeur finale. Enregistrer la valeur, soit pour la réutiliser dans une autre action, soit pour identifier l'itération actuelle.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.13.2. Utiliser un filtre de données

Cette action exécute les règles du filtre sur la source de données d'entrée. L'action va extraire les champs des données d'entrée et relier leurs valeurs aux variables associées.

Donc, l'action exécute le filtre sélectionné et assigne les valeurs respectives aux variables.

- **Éléments de niveau inférieur :** L'action peut créer des éléments de sous-niveau, identifiés par « **pour chaque ligne** » ou « **pour chaque bloc de données ...** ». S'il y en a, le filtre va extraire les données, non pas au niveau du document (avec des positions de champs codés en dur), mais au niveau des sous

ensembles qui contiennent des sections répétables. Dans ce cas, vérifier que les actions sont placées sous ces éléments. Il faut indenter l'action sous ce type d'élément.

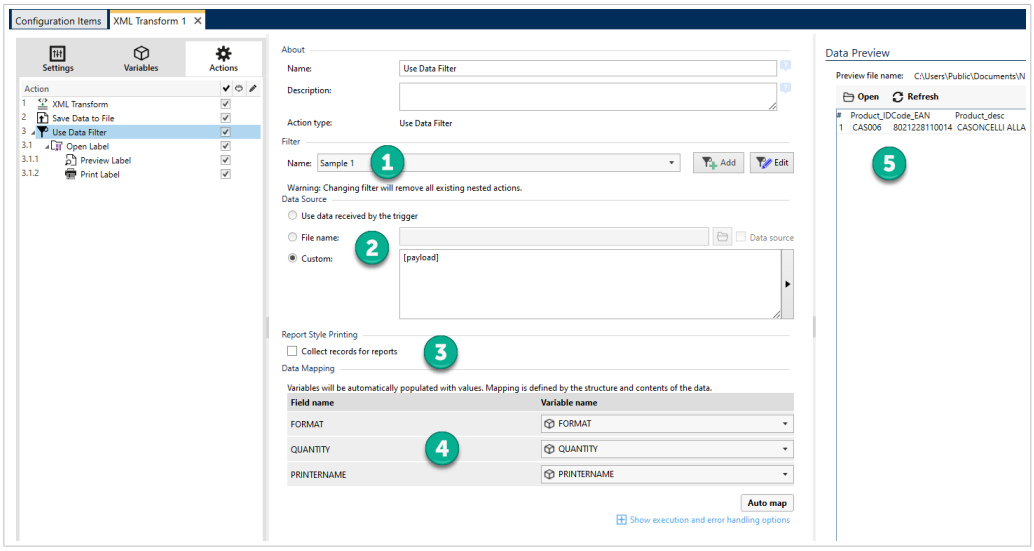
- **Mappage des champs avec les variables :** Le mappage entre les variables du déclencheur et les champs du filtre est soit manuel, soit automatique, selon la configuration du filtre. Quand les champs du filtre sont définis manuellement, il faut aussi les relier aux variables correspondantes manuellement.

> **NOTE**
> Il est conseillé de définir les champs en utilisant les mêmes noms que pour les variables de l'étiquette. Dans ce cas, le bouton **Mappage Auto** va relier automatiquement les noms correspondants.

- **Test de l'exécution du filtre :** Quand le mappage des champs avec les variables est terminé, il est possible de tester l'exécution du filtre. Le résultat s'affichera à l'écran dans une table. Le nombre de lignes dans la table représente le nombre de fois où les actions vont s'exécuter au niveau sélectionné. Les noms de colonnes représentent les noms de variables. La cellule contient les valeurs que le filtre va assigner à la variable correspondante. Le nom de fichier d'aperçu par défaut provient de la définition du filtre, le filtre peut s'exécuter sur tous les fichiers.

- **Collecter des enregistrements pour les rapports** collecte vos données afin que vous utilisiez des filtres de données pour créer des rapports. Pour plus d'informations, consulter l'article Section 9.7, « Automatisation des rapports ».

> **INFO NIVEAU DE PRODUIT**
> Cette fonctionnalité est disponible dans **LMS Enterprise**, **LMS Pro**, **Label Cloud Business** et **Label Cloud Compliance**.



Parcourir votre interface de filtre de données.

1. Choix du filtre

2. Les données que votre filtre utilise pour exécuter les règles.
3. Collecte d'enregistrements pour les rapports.
4. Mappage de vos champs (du filtre) aux variables (de l'étiquette/déclencheur).
5. Aperçu de l'exécution du filtre.

Pour plus d'informations, consulter la section Comprendre les Filtres et la section Exemples dans le guide utilisateur de NiceLabel Automation.

Le groupe **Filtre** permet de sélectionner le filtre à utiliser.

- **Nom :** Spécifie le nom du filtre à appliquer. Il peut être soit codé en dur, soit fourni dynamiquement par une variable nouvelle ou existante. La liste contient tous les filtres définis dans la configuration actuelle. Les trois derniers éléments de la liste permettent de créer un nouveau filtre.

> **NOTE**
> La sélection d'un autre filtre va enlever toutes les actions indentées sous cette action. Pour conserver les actions actuellement définies, les déplacer en dehors de l'action **Utiliser le filtre de données**. En cas de perte des actions, **Annuler** la dernière action pour revenir à la configuration précédente.

Le groupe **Source de données** permet de définir le contenu à envoyer à l'imprimante.

- **Utiliser les données reçues par le déclencheur :** Définit l'utilisation par le filtre des données reçues par le déclencheur. Dans ce cas, l'action va utiliser les données originales reçues par le déclencheur et exécuter les règles du filtre sur celles-ci.

> **Exemple**
>
> Par exemple, dans un déclencheur fichier, les données sont le contenu du fichier surveillé. Avec un déclencheur de base de données, les données font partie d'un jeu de données fournies par la base de données. Avec un déclencheur TCP/IP, les données sont le contenu brut reçu sur un socket.

- **Nom du fichier :** Définit le chemin et le nom du fichier contenant les données à filtrer. Le contenu du fichier spécifié est utilisé dans un filtre. L'option **Source de données** active le nom de fichier variable. Il faut sélectionner une variable qui contient le chemin et/ou le nom du fichier.
- **Personnalisé :** Définit un contenu personnalisé à analyser avec le filtre. Il peut être fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquer sur le bouton avec une flèche à droite de la zone de données et insérer la variable de la liste. Pour plus d'informations, consulter l'article Utilisation de Valeurs Composées dans le guide utilisateur de NiceLabel Automation.

Le champ **Aperçu des données** présente le résultat du filtre après lecture du contenu du fichier et application du filtre.

Les règles du filtre vont extraire les champs. La table va afficher le résultat de l'extraction. Chaque ligne de la table représente les données pour une étiquette. Chaque colonne représente une variable.

Pour voir le résultat, configurer le mappage des champs avec les variables correspondantes. En fonction de la définition du filtre, le mappage des variables aux champs se fait manuellement ou automatiquement.

- **Aperçu du nom de fichier :** Spécifie le fichier qui contient l'échantillon de données qui sera analysé dans le filtre. Le fichier d'aperçu est copié de la définition du filtre. Si le nom du fichier d'aperçu est changé, le nouveau nom de fichier sera enregistré.
- **Ouvrir :** Sélectionne un autre fichier sur lequel les règles du filtre vont s'appliquer.
- **Actualiser :** Relance le filtre sur le contenu du fichier d'aperçu. Le champ **Aperçu de Données** sera mis à jour avec le résultat.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes **`ActionLastErrorId`** et **`ActionLastErrorDesc`**.

5.13.3. Pour chaque enregistrement

Cette action exécute de multiple fois les actions indentées subordonnées. Toutes les actions indentées sont exécutées dans une boucle tant qu'il y a des enregistrements dans la table du formulaire connectée à une base de données.

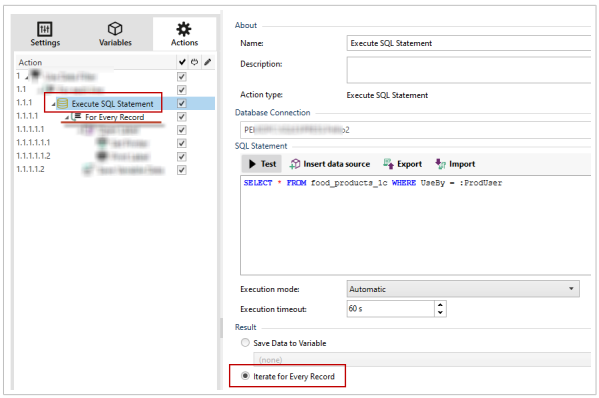
À propos: Ce groupe identifie l'action sélectionnée.

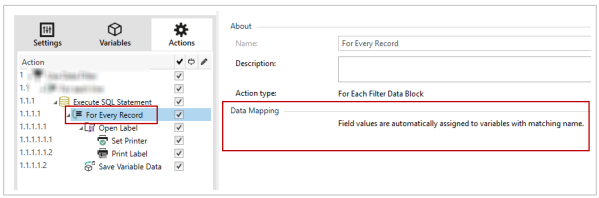
- Nom: Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- Description: Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- Type d'action: Information en lecture seule sur le type d'action sélectionné.

Le groupe **Paramètres** sélectionne les enregistrements.

- **Table du formulaire** : Table qui contient les enregistrements pour lesquels une action doit se répéter.
- **Utiliser tous les enregistrements** : répète une action pour tous les enregistrements d'une table donnée.
- **Utiliser l'enregistrement sélectionné** : répète une action uniquement pour les enregistrements sélectionnés.

Si vous utilisez l'action **Exécuter une requête SQL** avec l'option **Répéter pour chaque enregistrement** activée, NiceLabel insère automatiquement l'action **Pour chaque enregistrement**. Une note apparaît, à propos du mappage automatique de vos variables.





Les variables d'invite sur votre étiquette se connectent automatiquement à vos champs de base de données avec les mêmes noms. Voir les instructions sur la manière de créer vos solutions :

IMPORTANT

Lorsque vous créez votre solution avec une connexion de base de données, utilisez les variables d'invite sur vos étiquettes au lieu de champs de base de données.

Utilisez les mêmes noms pour les variables d'invite que ceux définis pour les champs de base de données, par exemple :

Champ de base de données : `food_products_1c.ProdCode`

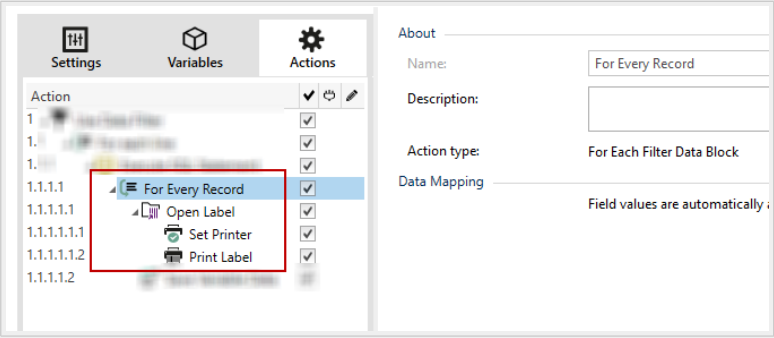
Variable d'invite sur votre étiquette : `ProdCode`

NiceLabel associe ensuite automatiquement les variables correspondantes aux champs de base de données.

Exemple

Votre solution est connectée à votre base de données. Votre configuration filtre les enregistrements de base de données requis et vous souhaitez imprimer les enregistrements filtrés sur vos étiquettes.

Utilisez l'action **Pour chaque enregistrement** et indentez les actions pour imprimer vos étiquettes.



Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (**`vrai`** ou **`faux`**). Quand le résultat de l'expression est **`vrai`**, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14. Données et connectivité

5.14.1. Ouvrir un Document / Programme

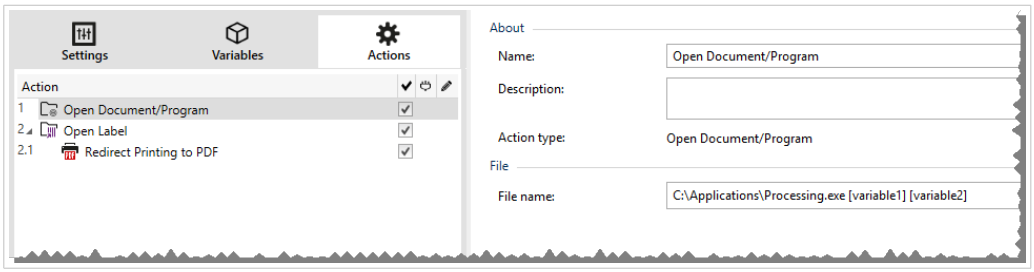
Cette action fournit une interface avec une application externe et l'ouvre en ligne de commande.

Les applications externes peuvent exécuter des fonctions additionnelles et renvoyer les résultats à NiceLabel 10. Cette action permet à NiceLabel Automation de se relier à un logiciel tiers qui peut traiter des données additionnelles, ou acquérir des données. Le logiciel externe peut fournir une réponse de données en l'enregistrant dans un fichier, duquel vous pouvez la consulter sous forme de variables.

> **NOTE**
> NiceLabel Automation fonctionne comme une application de service et ne peut pas interagir avec le bureau de l'utilisateur, même s'il fonctionne avec les privilèges de l'utilisateur actuellement connecté. Microsoft a empêché ce type d'interaction pour des raisons de sécurité.
>
> De ce fait, l'action **Ouvrir un document/programme** dans NiceLabel Automation ne peut ouvrir que des programmes qui s'exécutent en arrière-plan.

Vous pouvez fournir les valeurs de variable(s) au programme en les entourant de crochets dans la ligne de commande.

```
C:\Applications\Processing.exe [variable1] [variable2]
```



> **NOTE**
> Si vous utilisez cette action dans les solutions NiceLabel 10, elle vous permet d'ouvrir des pages Web ou de créer des messages électroniques directement à partir de vos formulaires. Voir la section Créer des liens hypertexte et envoyer des e-mails sur un formulaire dans le guide utilisateur de NiceLabel 10.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Fichier** définit le fichier à ouvrir.

- **Nom de fichier** : Définit le chemin et nom du fichier ou de l'application à ouvrir.
  Le nom et le chemin du fichier peuvent être codés en dur, et le même fichier sera utilisé à chaque fois. Si le nom du fichier est défini sans le chemin, le dossier comportant le fichier de configuration d'NiceLabel Automation (.MISX) sera utilisé. En utilisant une référence relative au nom de fichier, le dossier avec le fichier .MISX est utilisé comme dossier racine.
  Source de données: active le nom de fichier variable. Sélectionner une variable qui contient le chemin et/ou le nom du fichier, ou combiner plusieurs variables pour créer le nom du fichier. Pour plus d'informations, consulter l'article Utilisation des valeurs composées dans le guide utilisateur de NiceLabel Automation.

  > **NOTE**
  > Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

Le groupe **Options d'exécution** permet de paramétrer les détails d'ouverture du programme.

- **Attendre la fin** : Spécifie qu'il faut attendre la fin de cette action avant de continuer à exécuter les autres actions programmées en suivant.

> **ASTUCE**
> Activer cette option si l'action suivante dépend du résultat de l'application externe.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.2. Enregistrer les données dans un fichier

INFO NIVEAU DE PRODUIT

Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action enregistre la valeur variable ou autres flux de données (telles que les données binaires) dans le fichier. Le service NiceLabel Automation doit avoir les droits d'accès en écriture dans le dossier spécifié.

Le groupe **Fichier** définit le fichier à ouvrir.

- **Nom de fichier** : Emplacement du fichier ou du programme à ouvrir.
  Le nom et le chemin du fichier peuvent être codés en dur, et le même fichier sera utilisé à chaque fois. Si le nom du fichier est défini sans le chemin, le dossier comportant le fichier de configuration d'NiceLabel Automation (.MISX) sera utilisé. En utilisant une référence relative au nom de fichier, le dossier avec le fichier .MISX est utilisé comme dossier racine.
  Source de données : active le nom de fichier variable. Sélectionner une variable qui contient le chemin et/ou le nom du fichier, ou combiner plusieurs variables pour créer le nom du fichier. Pour plus d'informations, consulter l'article Utilisation des valeurs composées dans le guide utilisateur de NiceLabel Automation.

Le groupe **Si le fichier existe** gère les options en cas de fichier existant.

- **Écraser le fichier** : Écrase les données existantes avec les nouvelles données. L'ancien contenu est perdu.
- **Joindre les données au fichier** : Joint les valeurs de la variable au fichier de données existantes.

Le groupe **Contenu** définit les données à écrire dans le fichier spécifié.

- **Utiliser les données reçues par le déclencheur :** Les données reçues par le déclencheur seront enregistrées dans le fichier. En fait, cela réalisera une copie des données entrantes.
- **Personnalisé(e)** : Enregistre le contenu fourni dans le cadre du texte. Le contenu peut être un mélange de valeurs fixes, variables et caractères spéciaux. Pour insérer des variables et des caractères spéciaux, cliquer sur le bouton avec une flèche à droite de la zone de texte. Pour plus d'informations, consulter l'article Combinaison de valeurs dans un objet dans le guide utilisateur de NiceLabel Automation.
- **Encodage** : Spécifie l'encodage des données envoyées. **Auto** définit automatiquement l'encodage. Si nécessaire, sélectionner le type d'encodage préféré dans le menu déroulant.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.3. Lecture des données d'un fichier

> **INFO NIVEAU DE PRODUIT**
> La fonctionnalité décrite se trouve dans **NiceLabel LMS Enterprise** et **NiceLabel LMS Pro**.

Cette action lit le contenu du fichier fourni et l'enregistre dans la variable. Elle peut lire le contenu de tout type de fichiers, y compris les données binaires.

En principe, le module Automation Builder reçoit les données pour l'impression de l'étiquette avec le déclencheur. Par exemple : Avec le déclencheur fichier, le contenu du fichier déclencheur est automatiquement lu et analysé par des filtres. Mais il faut parfois contourner les filtres pour obtenir des

données externes. Après exécution de cette action et sauvegarde des données dans une variable, ces données sont de nouveau utilisables avec une des actions.

Cette action est utile :

- Pour combiner les données reçues par le déclencheur avec les données sauvegardées dans un fichier.

> **AVERTISSEMENT**
> Pour charger des données de fichiers binaires (comme des fichiers image bitmap ou d'impression), vérifier que la variable dans laquelle les données lues sont enregistrées est définie comme **variable binaire**.

- Pour échanger des données entre les déclencheurs. Un déclencheur prépare les données et les sauvegarde dans le fichier (en utilisant l'action Enregistrer les données dans un fichier), l'autre déclencheur lit les données.

Fichier : Nom du fichier dans lequel il faut lire les données.

- **Nom de fichier** : Emplacement du fichier ou du programme dans lequel cette action va lire les données.
  Le nom et le chemin du fichier peuvent être codés en dur, et le même fichier sera utilisé à chaque fois. Si le nom du fichier est défini sans le chemin, le dossier comportant le fichier de configuration d'NiceLabel Automation (.MISX) sera utilisé. En utilisant une référence relative au nom de fichier, le dossier avec le fichier .MISX est utilisé comme dossier racine.
  Source de données : active le nom de fichier variable. Sélectionner une variable qui contient le chemin et/ou le nom du fichier, ou combiner plusieurs variables pour créer le nom du fichier. Pour plus d'informations, consulter l'article Utilisation des valeurs composées dans le guide utilisateur de NiceLabel Automation.

  > **NOTE**
  > Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

Le groupe **Contenu** détermine les détails relatifs au contenu du fichier.

- **Variable** : Variable qui comporte le contenu du fichier. Il faut définir au moins une variable.
- **Encodage** : Spécifie l'encodage des données envoyées. **Auto** définit automatiquement l'encodage. Si nécessaire, sélectionner le type d'encodage préféré dans le menu déroulant.

  > **NOTE**
  > Il est impossible d'encoder des données si elles proviennent d'une variable binaire. Dans ce cas, la variable contiendra les données telles quelles.

Le groupe **Ressayer après échec** définit comment l'action peut continuer si le fichier spécifié devient inaccessible.

> **ASTUCE**
> Quand Automation Builder ne peut pas accéder au fichier, c'est peut être parce qu'il est verrouillé par une autre application. Si une application écrit encore des données dans le fichier et l'a bloqué en mode exclusif, aucune autre application ne peut l'ouvrir en même temps, même pas en lecture. Les causes de nouvelles tentatives sont les suivantes : le fichier n'existe pas (encore), le dossier n'existe pas (encore), l'utilisateur du service ne dispose pas des droits d'accès au fichier.

- **Nouvelles tentatives** : Spécifie le nombre de tentatives pour accéder au fichier. Si la valeur est 0, il n'y aura aucune tentative.
- **Intervalle entre les tentatives** : Spécifie l'intervalle de temps entre les essais défini en millisecondes.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions.

Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.4. Effacer un fichier

INFO NIVEAU DE PRODUIT

La fonctionnalité décrite se trouve dans **NiceLabel LMS Enterprise** et **NiceLabel LMS Pro**.

Cette action efface le fichier sélectionné dans un lecteur.

Le module NiceLabel Automation tourne en service sous un compte utilisateur Windows défini. Vérifier que le compte a les permissions pour effacer le fichier dans le dossier spécifié.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Fichier : Ce groupe paramètre les détails relatifs au fichier.

- **Nom du fichier** : Le nom du fichier à supprimer. Le **Nom du Fichier** peut être codé en dur. La **source de données** définit dynamiquement le **Nom du fichier** en utilisant une variable nouvelle ou existante.
  Le nom et le chemin du fichier peuvent être codés en dur, et le même fichier sera utilisé à chaque fois. Si le nom du fichier est défini sans le chemin, le dossier comportant le fichier de configuration d'NiceLabel Automation (.MISX) sera utilisé. En utilisant une référence relative au nom de fichier, le dossier avec le fichier .MISX est utilisé comme dossier racine.
  Source de données active le nom de fichier variable. Sélectionner ou créer une variable qui contient le chemin et/ou le nom du fichier, ou combiner plusieurs variables pour créer le nom du fichier. Pour plus d'informations, consulter l'article Utilisation des valeurs composées dans le guide utilisateur de NiceLabel Automation.

  NOTE

  Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.5. Exécuter une requête SQL

> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action envoie des commandes SQL sur un serveur SQL et collecte les résultats. Utiliser les commandes SELECT, INSERT, UPDATE et DELETE.

L'action Exécuter une requête SQL a deux objectifs :

- **Récupérer les données additionnelles dans une base de données :** Dans le module Automation Builder, un déclencheur reçoit les données d'impression, mais pas toutes les données requises. Par exemple, un déclencheur reçoit les données pour **`Product ID`** et **`Description`**, mais pas pour le **`Prix`**. Il faut rechercher la valeur du **`Prix`** dans la base de données SQL.

  Exemple de code SQL :

  ```
  SELECT Price FROM Products
  WHERE ID = :(Product ID)
  ```

  L'**`ID`** est un champ de la base de données, **`Product ID`** est une variable définie dans le déclencheur.

- **Mettre à jour ou supprimer les enregistrements de la base de données :** Après impression de l'étiquette, mettre à jour l'enregistrement de la base de données et envoyer un signal au système pour dire que cet enregistrement a déjà été traité.

  Exemple de code SQL :

  Changer le champ **`AlreadyPrinted`** (déjà imprimé) en **`True`** (vrai) pour l'enregistrement en cours de traitement.

  ```
  UPDATE Products
  SET AlreadyPrinted = True
  WHERE ID = :(Product ID)
  ```

  Ou effacer l'enregistrement actuel de la base de données, car il n'est plus nécessaire.

  ```
  DELETE FROM Products
  WHERE ID = :(Product ID)
  ```

  L'**`ID`** est un champ de la base de données, **`Product ID`** est une variable définie dans le déclencheur.

NOTE

Pour utiliser la valeur d'une variable dans une instruction SQL, il faut utiliser le signe deux points (:) devant son nom. Cela signale que le nom de la variable suit.

IMPORTANT

Lorsque vous créez votre solution avec une connexion de base de données, utilisez les variables d'invite sur vos étiquettes au lieu de champs de base de données.

Utilisez les mêmes noms pour les variables d'invite que ceux définis pour les champs de base de données, par exemple :

Champ de base de données : `food_products_1c.ProdCode`

Variable d'invite sur votre étiquette : `ProdCode`

NiceLabel associe ensuite automatiquement les variables correspondantes aux champs de base de données.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Connexion à la base de données** définit la connexion à la base de données qui sera utilisée pour l'instruction.

ASTUCE

Avant d'envoyer une requête SQL à une base de données, il faut paramétrer la connexion à la base de données. Cliquer sur le bouton **Définir** et suivre les instructions à l'écran. Pour se connecter à une source de données contrôlée par des requêtes SQL, il ne faut pas utiliser de fichiers texte (CSV) et Excel.

Le groupe **Instruction SQL** définit une instruction ou une requête SQL à exécuter.

ASTUCE

Les instructions en Langage de Manipulation des données (DML) peuvent exécuter des requêtes dans des tables de bases de données existantes.

Utiliser les instructions SQL standard, comme SELECT, INSERT, DELETE et UPDATE, y compris les jointures, fonctions et mots clés. Les instructions en langage DDL pour créer des bases de données et des tables (CREATE DATABASE, CREATE TABLE), ou les supprimer (DROP TABLE) ne sont pas permises.

- **Test** : Ouvre la section **Aperçu des données**. Simuler l'exécution (sélectionné par défaut) teste l'exécution des instructions SQL. Cliquer sur **Exécuter** pour lancer la simulation.

> **ASTUCE**
> **Aperçu des données** Cette section permet de tester l'exécution des requêtes SQL avec des données réelles. Pour protéger les données d'une mise à jour accidentelle, vérifier que l'option **Simuler l'exécution** est activée. Les instructions INSERT, DELETE et UPDATE s'exécuteront. Cela montrera en retour le nombre de données affectées, la transaction sera ensuite annulée.

Si la requête SQL utilise des variables de déclencheur, leurs valeurs peuvent être saisies pour l'exécution du test.

- **Insérer une source de données** : permet d'insérer une variable, nouvelle ou existante, dans une requête SQL.
- **Exporter/Importer** : Permet d'exporter ou importer une instruction SQL de ou dans un fichier externe.
- **Mode Exécution** : Spécifie le mode explicite d'exécution de la requête SQL.

> **ASTUCE**
> Avec certaines requêtes SQL complexes, il devient très difficile de déterminer automatiquement quelle est l'action prévue. Si la logique intégrée a des problèmes pour identifier vos intentions, sélectionner l'action principale manuellement.

  - **Automatique** : Détermine l'action automatiquement.
  - **Retourner un ensemble d'enregistrements (SELECT)** : reçoit les données récupérées dans les enregistrements.
  - **Ne renvoie pas l'ensemble d'enregistrements (INSERT, DELETE, UPDATE)** : Utiliser cette option si la requête ne renvoie pas les enregistrements. Ou insérer de nouveaux enregistrements, supprimer ou mettre à jour les enregistrements existants. Le résultat est une réponse sur le nombre de lignes affectées par votre requête.
- **Délai d'exécution** : vous permet de définir le délai pour d'envoi de vos commandes au serveur SQL. Utiliser le délai d'exécution dans le cas d'envoi de commandes SQL multiples et consécutives nécessitant un temps de traitement plus long.
  Saisir la durée du délai requis en secondes. Par défaut, la durée du délai d'exécution est de 60 s. Si vous voulez que votre fournisseur de base de données définisse le délai, saisir 0 s.

Le groupe **Résultat** permet de déterminer comment stocker le résultat de la requête SQL et de définir la répétition de l'action.

- **Enregistrer le résultat dans la variable** : définit la variable dans laquelle sera enregistré le résultat de l'instruction SQL. Cette option dépend du **Mode d'exécution** sélectionné.
  - **Résultat de la requête SELECT.** L'exécution de la requête SELECT renvoie un jeu d'enregistrements. Le contenu du texte reçu sera au format CSV. La première ligne contient les noms de champs de résultats. Les lignes suivantes contiennent les enregistrements.

> **NOTE**
> Pour extraire les valeurs des ensembles de données renvoyées et les utiliser dans d'autres actions, définir et exécuter l'action Utiliser le filtre de données sur le contenu de cette variable. Action disponible dans Automation Builder).

- **Résultat des requêtes INSERT, DELETE et UPDATE.** Les requêtes INSERT, DELETE et UPDATE renvoient un chiffre indiquant le nombre d'enregistrements affectés dans la table.

- **Répéter pour chaque enregistrement**. Si c'est activé, NiceLabelune nouvelle action **Pour chaque enregistrement s'ajoute automatiquement**. Voir plus d'informations sur cette action dans une rubrique dédiée.

> **NOTE**
> Le mappage automatique est activé. L'action Pour chaque enregistrement ne peut pas être supprimée.
>
> Champ de base de données : `food_products_1c.ProdCode`
>
> Variable d'invite sur votre étiquette : `ProdCode`

Réessayer après échec : Ce groupe permet de configurer l'action pour réessayer continuellement de rétablir la connexion à la base de données si la première tentative n'a pas réussi. Si l'action échoue au cours du nombre de tentatives défini, une erreur sera signalée.

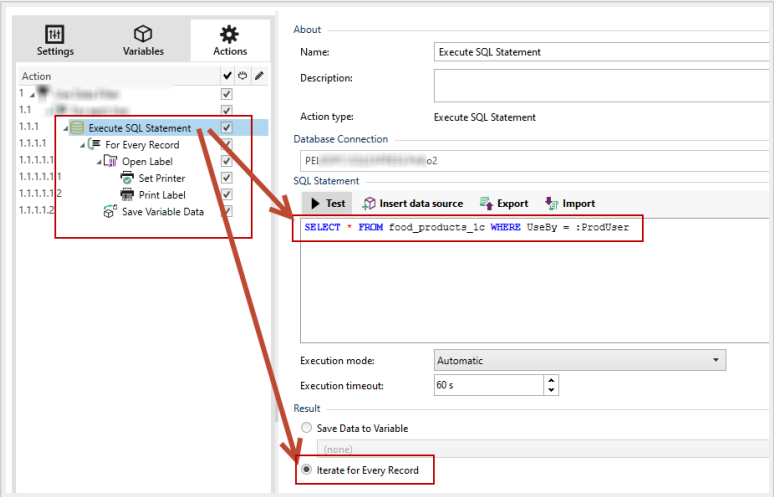
- **Nouvelles tentatives** : Spécifie le nombre de tentatives de connexion au serveur de la base de données.
- **Intervalle entre les tentatives** : Spécifie le temps d'attente entre chaque essai.

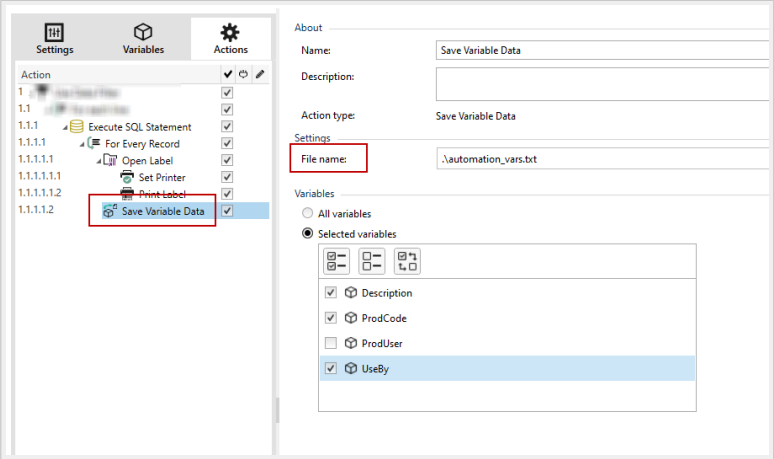
Exemple

Vous souhaitez imprimer des étiquettes avec des données de votre base de données `food_products_1c`, mais uniquement des enregistrements avec une valeur de champ prédéfinie `UseBy` (date limite de consommation).

Vous définissez la valeur de `UseBy` avec la variable `ProdUser`. Dans ce cas, la valeur **ProdUser** est « 3 ».

Après l'impression de chaque étiquette, NiceLabel écrit les valeurs de base de données dans un fichier texte sur votre disque. Utilisez la chaîne suivante :





Seules les étiquettes où la valeur `UseBy` est égale à « 3 » sont imprimées et les valeurs sont écrites dans un fichier texte :

```
"Description";"ProdCode";"UseBy"
"Choux pastry";"977000000011";"3"
"Croline";"977000000012";"3"
"Danish pastry";"977000000013";"3"
"Kouign-amann";"977000000022";"3"
"Muskazine";"977000000026";"3"
"Pineapple bun";"977000000029";"3"
"Schneeball";"977000000033";"3"
"Schuxen";"977000000034";"3"
"Tortell";"977000000041";"3"
```

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.

- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.6. Envoyer les données au port TCP/IP

Envoie les données à tout périphérique acceptant une connexion TCP/IP sur un numéro de port prédéfini.

Envoyer les données au port TCP/IP établit la connexion avec le périphérique, envoie les données et termine la connexion. La connexion et la communication sont gérées par le protocole de communication qui s'établit entre le client et le serveur au début et à la fin de la connexion TCP.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Paramètres de connexion** permet de choisir les paramètres de connexion.

- **Répondre à l'expéditeur**: Permet de répondre directement au socket à partir duquel proviennent les données du déclencheur. Utiliser cette option pour renvoyer des informations sur le processus d'impression.

  > **NOTE**
  > Cette option est disponible dans NiceLabel Automation.

  Les conditions préalables pour le paramètre **Répondre à l'expéditeur** sont :

  - L'interlocuteur ne ferme pas le canal de communication une fois le message livré.
  - **Envoyer les Données au port TCP/IP** est utilisée dans le déclencheur du **Serveur TCP/IP**.
  - Ne **pas** configurer l'événement d'exécution dans le déclencheur du **serveur TCP/IP À la déconnexion du client**.

- **Destination (adresse IP:port)** : Définit l'adresse et le port de destination du serveur TCP/IP. Coder en dur les paramètres de connexion et utiliser une adresse IP fixe ou utiliser une variable en cliquant sur la flèche à droite et en sélectionnant la variable prédéfinie. Pour plus d'informations, consulter l'article Combinaison de valeurs dans un objet dans le guide utilisateur de NiceLabel Automation.

> **Exemple**
>
> Si la variable **`hostname`** procure le nom du serveur TCP/IP et la variable **`port`** fournit le numéro de port, saisir la destination suivante :
>
> `[hostname]:[port]`

- **Délai de déconnexion** : Prolonge la connexion sur le socket ciblé de l'intervalle de temps défini après que les données ont été fournies. Certains périphériques ont besoin de plus de temps pour traiter les données. Taper le différé à la main ou cliquer sur les flèches pour augmenter ou diminuer la valeur.
- **Enregistrer la réponse de données dans une variable** : Permet de créer ou sélectionner une variable où sont enregistrées les données reçues du serveur. Toutes les données reçues du serveur TCP/IP après le « délai de déconnexion » sont enregistrées dans la variable.

Le groupe **Contenu** définit le contenu à envoyer au serveur TCP/IP.

> **ASTUCE**
> Utiliser le contenu fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquer sur le bouton avec une flèche à droite de la zone de données et insérer la variable de la liste. Pour plus d'informations, consulter l'article Combinaison de valeurs dans un objet dans le guide utilisateur de NiceLabel Automation.

- **Données** : Spécifie le contenu qui sera envoyé en sortie.
- **Encodage** : Spécifie l'encodage des données envoyées. Auto définit automatiquement l'encodage. Si nécessaire, sélectionner le type d'encodage préféré dans le menu déroulant.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.7. Envoyer les données au port série

Cette action envoie les donnéessur un port série. Cette action permet de communiquer avec les périphériques connectés à un port série.

ASTUCE

Il faut que la configuration du port série soit identique des deux côtés, dans l'action et sur le périphérique en série. Le port série peut être utilisé par une application dans la machine. Pour que cette action puisse utiliser ce port, aucune autre application ne doit l'utiliser, même pas un pilote d'imprimante.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Port** permet de choisir le port série.

- **Nom du port** : port sur lequel le périphérique est connecté. Cela peut être un port COM matériel ou un port COM virtuel.

Le groupe **Paramètres de port** permet de définir d'autres paramètres pour se connecter au port série.

- **Bits par seconde** : Vitesse utilisée par un périphérique pour communiquer avec le PC. L'alias généralement utilisé pour ce paramètre est « baud rate ». Sélectionner la valeur dans la liste du menu déroulant.
- **Bits de données** : Spécifie le nombre de bits de données dans chaque caractère. 8 bits de données sont généralement utilisés dans les appareils récents. Sélectionner la valeur dans la liste du menu déroulant.
- **Parité** : Spécifie la méthode de détection d'erreurs de transmission. Le paramètre de parité généralement utilisé est « aucune », avec la détection d'erreur gérée par un protocole de communication (contrôle de flux). Sélectionner la valeur dans la liste du menu déroulant.
- **Bits d'arrêt** : Les bits d'arrêt envoyés à la fin de chaque caractère permettent à la machine de réception de détecter la fin d'un caractère et de le resynchroniser avec le flux de caractères. Les appareils électroniques utilisent généralement un bit de stop. Sélectionner la valeur dans la liste du menu déroulant.
- **Contrôle de flux** : Le port série peut utiliser les signaux de l'interface pour interrompre et reprendre la transmission des données.

Le groupe **Contenu** définit le contenu à envoyer au port série.

> **ASTUCE**
> Il peut être fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquer sur le bouton avec une flèche à droite de la zone de données et insérer la variable de la liste. Pour plus d'informations, consulter l'article Combinaison de valeurs dans un objet dans le guide utilisateur de NiceLabel Automation.

- **Données** : Spécifie le contenu qui sera envoyé en sortie.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (**`vrai`** ou **`faux`**). Quand le résultat de l'expression est **`vrai`**, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

  > **NOTE**
  > Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes **`ActionLastErrorId`** et **`ActionLastErrorDesc`**.

5.14.8. Lecture des données sur le port série

Cette action collecte les données reçues par le port série (RS-232) et les enregistre dans une variable sélectionnée. Cette action permet de communiquer avec les périphériques connectés à un port série.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Port** permet de choisir le port série.

- **Nom du port** : port sur lequel le périphérique est connecté. Cela peut être un port COM matériel ou un port COM virtuel.

Le groupe **Paramètres de port** permet de définir d'autres paramètres pour se connecter au port série.

- **Bits par seconde** : Vitesse utilisée par un périphérique pour communiquer avec le PC. L'alias généralement utilisé pour ce paramètre est « baud rate ».
- **Bits de données** : Spécifie le nombre de bits de données dans chaque caractère. 8 bits de données sont généralement utilisés dans les appareils récents.
- **Parité** : Spécifie la méthode de détection d'erreurs de transmission. Le paramètre de parité généralement utilisé est « aucune », avec la détection d'erreur gérée par un protocole de communication (contrôle de flux).
- **Bits d'arrêt** : Les bits d'arrêt envoyés à la fin de chaque caractère permettent à la machine de réception de détecter la fin d'un caractère et de le resynchroniser avec le flux de caractères. Les appareils électroniques utilisent généralement un bit de stop.
- **Contrôle de flux** : Le port série peut utiliser les signaux de l'interface pour interrompre et reprendre la transmission des données.

Exemple

Un appareil lent peut avoir besoin de garder le contact avec le port série pour indiquer que les données doivent être mises en pause pendant qu'il traite les données reçues.

Le groupe **Options** comporte les paramètres suivants :

- **Différé de lecture** : Spécifie un différé éventuel durant la lecture de données sur le port série. Le contenu complet du buffer du port série sera lu à la fin du différé. Taper le différé à la main ou cliquer sur les flèches pour augmenter ou diminuer la valeur.

- **Envoyer les données d'initialisation** : Spécifie la chaîne de caractères qui est envoyée au port série sélectionné avant la lecture des données. Cette option permet d'initialiser le périphérique pour qu'il puisse fournir les données. L'utiliser aussi pour envoyer une question spécifique à l'appareil, et recevoir la réponse spécifique. Cliquer sur le bouton avec une flèche pour insérer des caractères spéciaux.

Le groupe **Extraction des données** permet de définir l'extraction de certaines parties des données reçues.

- **Point de départ** : Position de départ pour l'extraction des données.
- **Position de fin** : Position finale pour l'extraction des données.

Le groupe **Résultat** définit une variable pour le stockage des données.

- **Enregistrer les données dans une variable** : Sélectionner ou créer une variable pour y stocker les données reçues.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions.

Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.9. Envoyer les données à l'imprimante

Cette action envoie les données à l'imprimante sélectionnée. Cette action est utile pour envoyer des flux d'impression pré-générés à toute imprimante disponible.

NiceLabel Automation utilise les pilotes d'imprimante en mode de transit, pour pouvoir envoyer les données au port de destination, tel que le port LPT, COM, TCP/IP ou USB, sur lequel l'imprimante est connectée.

NOTE

Scénario possible. Les données reçues par le déclencheur doivent être imprimées sur la même imprimante réseau mais sur des masques d'étiquettes différents (fichiers .NLBL). L'imprimante peut accepter des données de différents postes de travail. Elle imprime généralement les travaux dans l'ordre reçu.Automation Builder va envoyer chaque masque d'étiquette dans des travaux d'impression distincts, donnant la possibilité à d'autres postes de travail d'insérer leurs impressions entre celles créées par Automation Builder. Au lieu d'envoyer chaque travail séparément à l'imprimante, il est possible de les fusionner (en utilisant l'action Rediriger l'impression vers un fichier) vers un fichier pour envoyer ensuite un seul gros travail d'impression à l'imprimante.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Imprimante** sélectionne l'imprimante.

- **Nom de l'imprimante** : Nom de l'imprimante à laquelle les données sont envoyées. Sélectionner l'imprimante dans la liste d'imprimantes installées localement, saisir un nom d'imprimante ou la définir dynamiquement avec une variable nouvelle ou existante.

Source de données définit le contenu à envoyer à l'imprimante.

- **Utiliser les données reçues par le déclencheur :** Définit l'utilisation des données reçues par le déclencheur. Dans ce cas, le flux d'imprimante reçu est utilisé en entrée dans le filtre. L'objectif est

de le rediriger vers l'imprimante sans qu'il soit modifié. Le même résultat peut être atteint en activant la variable interne **`DataFileName`** et en utilisant le contenu du fichier auquel elle se réfère. Pour plus d'informations, consulter l'article Utilisation de Valeurs Composées dans le guide utilisateur de NiceLabel Automation.

- **Nom de fichier** : Définit le chemin et nom de fichier contenant le flux d'impression. Le contenu du fichier spécifié est envoyé à l'imprimante. Sélectionner la **Source de données** pour définir dynamiquement le nom du fichier en utilisant une valeur variable.
- **Variable** : Définit la variable (nouvelle ou existante) qui contient le flux d'impression.
- **Personnalisé** : Définit un contenu personnalisé à envoyer à une imprimante. Il peut être fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquer sur le bouton avec une flèche à droite de la zone de données et insérer la variable de la liste. Pour plus d'informations, consulter l'article Combinaison de valeurs dans un objet dans le guide utilisateur de NiceLabel 10.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (**`vrai`** ou **`faux`**). Quand le résultat de l'expression est **`vrai`**, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions.

Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.10. Requête HTTP

INFO NIVEAU DE PRODUIT

Les fonctions Automation Builder nécessitent d'utiliser **LMS Enterprise**.

Cette action envoie les données au serveur Web de destination en utilisant la méthode HTTP sélectionnée. Les schémas d'URI HTTP et HTTPS sont autorisés.

HTTP fonctionne comme un protocole de requête-réponse entre client et serveur. Dans cette action, NiceLabel 10 fonctionne comme un client qui communique avec un serveur à distance. Cette action va soumettre la requête HTTP sélectionnée au serveur. Le serveur renverra un message de réponse, concernant l'état d'achèvement de la requête et la réponse dans le corps du message.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Paramètres de connexion** permet de choisir les paramètres de connexion.

NOTE

Cette action est compatible avec le protocole Internet version 6 (IPv6).

- **Destination** : L'adresse, le port et la destination (chemin) sur le serveur Web.

  ASTUCE

  La définition du port est facultative quand le serveur Web fonctionne sur le port 80 par défaut. Coder en dur les paramètres de connexion ou utiliser un nom d'hôte et une adresse IP fixes. Utiliser une valeur variable pour définir cette option dynamiquement. Pour plus d'informations, consulter l'article Utilisation des valeurs composées dans le guide utilisateur de NiceLabel Automation.

Exemple

Si la variable **`hostname`** procure le nom du serveur Web et la variable **`port`** fournit le numéro de port, saisir la destination suivante :

```
[hostname]:[port]
```

- **Méthode requise** : Affiche les méthodes de requête disponibles.
- **Délai écoulé** : Le temps imparti (en ms) pour que la connexion au serveur soit établie et la réponse reçue.
- **Enregistrer la réponse d'état dans une variable** : Définit la variable dans laquelle est enregistré le code d'état renvoyé par le serveur.

  ASTUCE
  L'état du code doit être de l'ordre de 2XX. Par exemple, 200 est une bonne réponse. Les codes 5XX sont des erreurs du serveur.

- **Enregistrer la réponse de données dans une variable** : Variable où sont enregistrées les données reçues du serveur.

Le groupe **Authentification** permet de sécuriser la connexion au serveur Web.

- **Activer une authentification de base** : Permet de mettre les identifiants nécessaires à la connexion au serveur Web. L'identifiant et le mot de passe peuvent être fixes ou fournis par une valeur variable.

NOTE
L'authentification Basique HTTP (BA) utilise les entêtes statiques standard HTTP. Le mécanisme BA ne procure aucune protection de confidentialité pour les infos d'identification transmises. Elles sont seulement encodées en Base64 pendant le transit, mais ne sont ni cryptées ni hachées. L'Authentification de Base devrait être utilisée en HTTPS.

- **Afficher le mot de passe** : Démasque les caractères de l'identifiant et du mot de passe.

Contenu permet de définir le contenu à envoyer au serveur Web.

- **Données** : Spécifie le contenu qui sera envoyé en sortie. Il peut être fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquer sur le bouton avec une flèche à droite de la zone de données et insérer la variable de la liste. Pour plus d'informations, consulter l'article Combinaison de valeurs dans un objet dans le guide utilisateur de NiceLabel 10.
- **Encodage** : Spécifie l'encodage des données envoyées.

> **ASTUCE**
> **Auto** définit automatiquement l'encodage. Si nécessaire, sélectionner le type d'encodage préféré dans le menu déroulant.

- **Type** : Spécifie la caractéristique Content-Type pour le message HTTP. Si aucun type n'est sélectionné, la valeur par défaut **`application/x-www-form-urlencoded`** est utilisée. Si la liste ne contient aucun type utilisable, il est possible d'en définir un personnalisé, ou déterminer une variable qui le définira dynamiquement..

Des **Entêtes HTTP supplémentaires** sont exigées par certain serveurs HTTP (spécialement pour les services REST).

- **Entêtes additionnelles** : Entêtes codées en dur ou récupérées d'une variable. Pour accéder aux variables, cliquer sur la petite flèche à la droite du champ de texte. Pour plus d'informations, consulter l'article Combinaison de valeurs dans un objet dans le guide utilisateur de NiceLabel 10.
  Certains serveurs HTTP (spécialement pour les services REST) obligent à inclure l'entête HTTP personnalisée dans le message. Cette section permet de fournir les entêtes HTTP nécessaires. L'entête HTTP doit être introduite en respectant la syntaxe suivante :

  ```
  header field name: header field value
  ```

  Par exemple, pour utiliser les noms des champs **`Accept`**, **`User-Agent`** et **`Content-Type`**, utiliser la syntaxe suivante :

  ```
  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
  ```

  Vous pouvez coder en dur les noms des champs des entêtes ou obtenir leurs valeurs à partir des variables des déclencheurs. Il est possible d'utiliser autant de champs d'entête personnalisés que nécessaires, mais chaque entête doit être placée sur une nouvelle ligne.

  > **NOTE**
  > Les entêtes HTTP saisies vont remplacer les entêtes déjà définies dans les propriétés des actions, telles que **Content-Type** (type de contenu).

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.

- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.14.11. Service Web

> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent d'utiliser **LMS Enterprise**.

Le Service Web est une méthode de communication entre deux appareils électroniques ou logiciels. Il est défini comme norme d'échange de données. Il utilise XML pour taguer les données, SOAP pour transférer les données et WSDL pour décrire les services disponibles.

Cette action se connecte à un Web service distant et exécute les méthodes sur celui-ci. Les méthodes sont comme des actions publiées sur le service Web. L'action va envoyer des valeurs à la méthode sélectionnée dans le service Web distant, collecter le résultat et l'enregistrer dans les variables sélectionnées.

Après avoir importé le WSDL et ajouté une référence au service Web, ses méthodes s'affichent dans la liste déroulante Méthode.

NOTE

Elle permet de transférer des données simples, telles que des chaînes de caractères, entières, booléennes, mais pas de données complexes. Le WSDL ne doit contenir qu'un seul lien.

NOTE

Il faut imprimer des étiquettes de produit. Le déclencheur ne doit recevoir que le segment de données nécessaire. Par example, le déclencheur reçoit la valeur pour **`ID Produit`** et **`Description`**, mais pas le **`Prix`**. L'information de prix est disponible dans une base de données séparée qui est accessible sur appel du Web Service. Le service Web définit la fonction utilisant une définition WSDL. Par exemple, la fonction entrée est **`Product ID`** et celle en sortie est **`Prix`**. L'action service Web envoie **`Product ID`** au service Web. Elle exécute une recherche dans la base de données et donne le **`Prix`** correspondant en retour. L'action enregistre le résultat dans une variable, qui peut être utilisée sur l'étiquette.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Définition du service web** comporte les paramètres suivants :

NOTE

Cette action est compatible avec le protocole Internet version 6 (IPv6).

- **WSDL** : Emplacement de la définition WDSL.
  Le WSDL est généralement fourni par le service Web. Il suffit de saisir le lien vers le WSDL et de cliquer sur le bouton **Importer** pour lire la définition. Si le WSDL est difficile à récupérer, enregistrer le WSDL dans un fichier et entrer le chemin et le nom du fichier pour en charger les méthodes. NiceLabel 10 détectera automatiquement si le service Web distant utilise la syntaxe document ou RPC et communique de façon appropriée.
- **Adresse** : Donne l'adresse du Service Web.
  Initialement, cette information est extraite de WSDL, mais elle peut être actualisée avant l'exécution de l'action. C'est utile pour les environnements de développement / test / production, qui utilisent la même liste d'actions, mais avec des noms différents de serveurs sur lesquels fonctionne le service Web.

Il peut être fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer un contenu de variable, cliquer sur le bouton flèche à droite de la zone de données et insérer la variable de la liste. Pour plus d'informations, consulter l'article Combinaison de valeurs dans un objet dans le guide utilisateur de NiceLabel 10.

- **Méthodes** : Liste les méthodes (fonctions) disponibles sur le service Web sélectionné. La liste est mise à jour automatiquement par la définition WSDL.
- **Paramètres** : Définit les variables d'entrée et de sortie pour la méthode (fonction) sélectionnée. Les paramètres d'entrée attendent une entrée. Pour détecter des erreurs et tester, saisir des valeurs fixes et visualiser le résultat à l'écran. ou sélectionner une variable. La valeur de cette variable sera utilisée comme paramètre d'entrée. Le paramètre de sortie fournit le résultat de la fonction. il faut sélectionner la variable qui va stocker le résultat.
- **Délai écoulé** : Le temps imparti (en ms) pour que la connexion au serveur soit établie.

Authentification Active l'authentification de base. Cette option définit les infos d'identification nécessaires à établir l'appel externe vers le service Web distant.

- **Activer l'authentification de base** : active la définition de l'**identifiant** et le **mot de passe** pouvant être tapé à la main ou définit par une variable. Sélectionner **Sources de données** pour sélectionner ou créer les variables.
- **Afficher le mot de passe** : démasque les caractères de l'**identifiant** et du **mot de passe** .
  Les détails concernant la sécurité se trouvent au chapitre Sécuriser l'accès aux déclencheurs dans le guide utilisateur de NiceLabel Automation.

Le champ **Aperçu des données** permet de tester l'exécution du service Web.

- Le bouton **Exécuter** exécute l'appel du service Web.
  Il envoie les valeurs des paramètres d'entrée au Web Service et fournit le résultat dans le paramètre de sortie. Utiliser cette fonctionnalité pour tester l'exécution du Web Service. Donner des valeurs aux paramètres de sortie et visualiser le résultat à l'écran. Si le résultat est satisfaisant, remplacer les valeurs fixes introduites du paramètre d'entrée par une variable de la liste.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.15. Autre

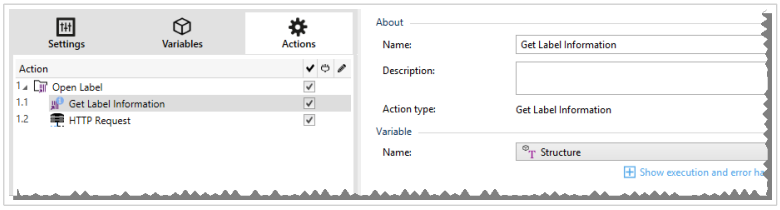
5.15.1. Récupérer les informations de l'étiquette

> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent la licence **LMS Pro**, **LMS Enterprise** ou **NiceLabel Cloud Business**.

Cette action renvoie les informations structurelles du fichier d'étiquette concerné. L'action fournit les informations concernant les dimensions de l'étiquette, le pilote d'imprimante et la liste de toutes les variables avec leurs propriétés principales.

L'action Récupérer les informations de l'étiquette renvoie les informations d'origine telles qu'enregistrées dans le fichier de l'étiquette. De plus, elle fournit les informations après simulation du processus d'impression. La simulation garantit que toutes les variables d'étiquette ont reçu une valeur comme c'est le cas durant une impression normale. Également les informations de hauteur de l'étiquette pour contrôler les dimensions correctes dans le cas où l'étiquette est de hauteur variable (dans ce cas, les dimensions de l'étiquette dépendent de la quantité de données à imprimer). L'action renvoie les dimensions de l'étiquette, pas les dimensions de page.

L'action enregistre les informations concernant la structure de l'étiquette dans une variable sélectionnée. Ces données peuvent être renvoyées au système en utilisant l'action Requête HTTP (ou une action de connexion externe similaire), ou dans la réponse du déclencheur, si vous utilisez un déclencheur bidirectionnel.



> **NOTE**
> Cette action doit être indentée sous l'action Ouvrir une étiquette.

Le groupe **Variable** sélectionne ou crée une variable pour stocker les informations concernant la structure de l'étiquette.

- **Nom :** Spécifie le nom de la variable. Il faut sélectionner ou créer une variable dans laquelle les informations de l'étiquette sont sauvegardées en format XML.
  - Si vous voulez utiliser les informations du XML à l'intérieur du déclencheur, vous pouvez le définir et l'exécuter avec l'action Utiliser un filtre de données (Automation Builder uniquement).
  - Si vous voulez retourner les données XML en tant que réponse dans votre déclencheur HTTP ou Web Service, utiliser cette variable directement dans le champ **Données de réponse** de la page de configuration du déclencheur.
  - Si vous voulez enregistrer les données XML dans un fichier, utiliser l'action Enregistrer les données dans un fichier.

Le groupe **Paramètres additionnels** permet d'utiliser des valeurs provisoires.

- **Utiliser des valeurs provisoires :** Remplace les valeurs des données manquantes par des valeurs provisoires.

> **ASTUCE**
> Pour plus de renseignements sur les valeurs provisoires, consulter la section Variables dans le guide utilisateur de NiceLabel 10 Designer.

Exemple d'Informations XML de l'étiquette

Cet exemple présente une vue structurelle des éléments et leurs attributs tels qu'ils sont renvoyés.

```
<?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>
```

Spécification des informations XML de l'étiquette

Cette section contient la description de la structure du fichier XML telle qu'elle est renvoyée par l'action Récupérer les informations de l'étiquette.

> **NOTE**
> Toutes les unités de mesure sont exprimées en 1/1000 mm. Par exemple, une largeur de 25000 correspond à 25 mm.

- **<Label> :** : Élément racine.
- **<Original> :** Spécifie les dimensions de l'étiquette et le nom d'imprimante tels qu'ils sont sauvegardés dans le fichier d'étiquette.
  - **Largeur :** Cet élément contient la largeur originale de l'étiquette.
  - **Hauteur :** Cet élément contient la hauteur originale de l'étiquette.
  - **PrinterName :** Cet élément contient le nom de l'imprimante pour laquelle l'étiquette a été créée.
- **Current :** Spécifie les dimensions de l'étiquette et le nom de l'imprimante après la simulation d'impression.

- **Largeur :** Cet élément contient la largeur actuelle de l'étiquette.
- **Hauteur :** Cet élément contient la hauteur actuelle de l'étiquette. Si l'étiquette est de hauteur variable, elle s'adapte en fonction de la taille des objets. Par exemple, les objets Paragraphe et RTF peuvent s'agrandir en hauteur entraînant la modification de la taille de l'étiquette.
- **PrinterName :** Cet élément contient le nom de l'imprimante à utiliser pour l'impression.

Exemple

Une imprimante autre que celle d'origine va être utilisée si le pilote d'imprimante d'origine n'est pas installé sur cet ordinateur, ou si l'imprimante a été modifiée avec l'action Définir l'imprimante.

- **<Variables> et <Variable> :** L'élément `Variables` contient la liste des variables saisies de l'étiquette, chacune étant définie dans un élément `Variable` distinct. Les variables de saisie se trouvent listées dans la boîte de dialogue Imprimer quand l'impression est effectuée depuis NiceLabel 10. S'il n'y a pas de variable saisie dans l'étiquette, l'élément `Variables` est vide.
  - **Nom :** Contient le nom de la variable.
  - **Description :** Contient la description de la variable.
  - **DefaultValue :** Contient les valeurs par défaut telles qu'elles ont été définies pour la variable lors du processus de création d'étiquette.
  - **Format :** Contient le type de contenu de la variable acceptable (caractères).
  - **IsPrompted :** Indique si la variable est saisie au moment de l'impression ou non.
  - **PromptTex t:** Contient le texte d'invite pour que l'utilisateur entre la valeur.
  - **CurrentValue :** Contient la valeur actuelle telle qu'utilisée pour l'impression.
  - **IncrementType :** Indique si la variable est définie en tant que compteur ou non. Si c'est un compteur, le type de compteur est précisé.
  - **IncrementStep :** Contient des informations concernant l'incrément du compteur. La valeur du compteur s'incrémente/décrémente de cette valeur sur l'étiquette suivante.
  - **IncrementCount :** Contient des informations concernant le point de la valeur du compteur qui s'incrémente/décrémente. Généralement, le compteur change de valeur à chaque étiquette, mais cela peut être modifié.
  - **Longueur :** Contient le nombre maximum de caractères stockés dans une variable.
  - **IsPickListEnabled :** Contient des informations qui permettent de savoir si l'utilisateur sélectionne ou non les valeurs de variable parmi une liste de choix.
  - **PickListValues :** Contient les valeurs actuelles de la liste de choix (sélectionnables).

Définition du Schéma XML (XSD) pour la Spécification XML de l'étiquette

```
<?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>
```

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

NOTE

Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

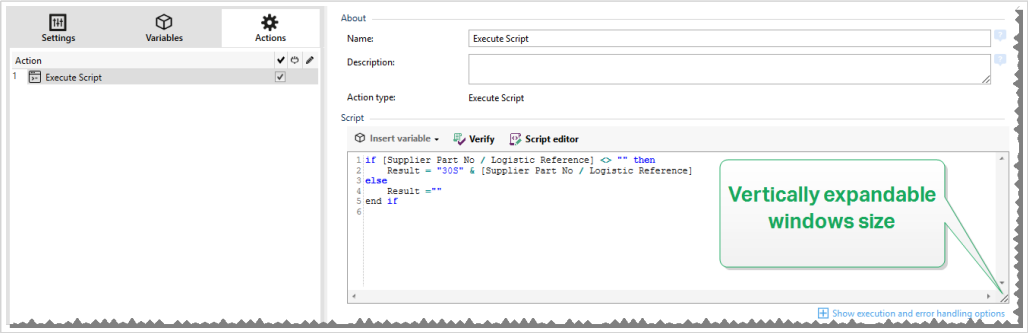
- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.15.2. Exécuter un script

L'utilisation de scripts VBScript ou Python personnalisés améliore les fonctionnalités du logiciel. Utiliser cette fonction si les actions prédéfinies ne sont pas suffisantes.

Les scripts peuvent inclure les variables des déclencheurs, variables internes ou variables définies ou importées des étiquettes.

Veiller à ce que le compte Windows, sous lequel le service tourne, dispose des droits pour exécuter les commandes du script.



> **NOTE**
> Le langage de script est configuré par déclencheur dans les propriétés du déclencheur. Dans un déclencheur, toutes les actions Exécuter le script doivent utiliser le même langage de script.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.

- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

L'éditeur de **Script** propose les fonctionnalités suivantes :

- **Insérer une source de données** : Permet d'insérer une variable, nouvelle ou existante, dans un script.
- **Vérifier** valide la syntaxe du script saisi.
- **Éditeur de script :** Ouvre l'éditeur qui rend l'écriture du script plus facile et plus efficace.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

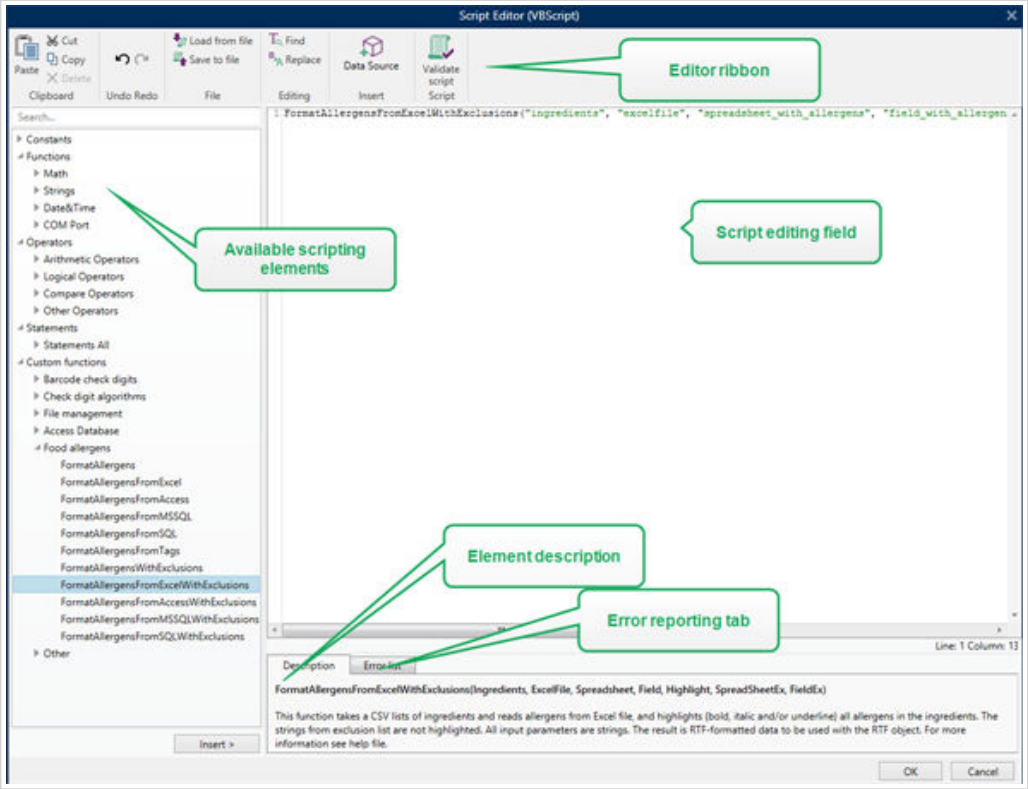
Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.15.2.1. Éditeur de script

NiceLabel 10 comporte un éditeur de scripts qui facilite l'écriture en Python ou en VB, sans erreur ni perte de temps.



La sélection des langages de script devant être utilisés dans l'Éditeur de script diffère entre NiceLabel Designer Pro et Automation Builder :

- Dans Designer, double-cliquez sur le plan de travail du formulaire pour ouvrir Propriétés du formulaire > Paramètres additionnels > Langue de script du formulaire.
- Dans Automation Builder, allez dans Éléments de configuration > cliquez sur Modifier pour ouvrir les propriétés du déclencheur > Paramètres > Autre > Script.

NiceLabel 10 Utilise la variante .NET de Python appelée IronPython. C'est une langue de script Python tout à fait compatible qui prend également en charge les méthodes .NET.

Le ruban de l'éditeur dispose des commandes usuelles qui sont réparties dans plusieurs groupes par fonction.

- Le groupe Presse-papiers comporte les commandes Couper, Copier, Coller et Supprimer.
- Le groupe de ruban Annuler Rétablir annule ou répète les actions d'édition.
- Le groupe Fichier permet de charger et enregistrer les scripts dans un fichier.
    - Charger depuis un fichier : charge un script à partir d'un fichier texte externe précédemment enregistré.

    - Enregistrer sur un fichier : Stocke le script actuellement modifié dans un fichier texte.
- Le groupe Édition permet de rechercher et remplacer des chaînes dans un script.
    - Chercher : Trouve la chaîne saisie dans le script.
    - Remplacer : Remplace la chaîne dans le script.
- Le groupe Insérer : La commande Source de données insère dans le script des sources de données existantes ou nouvellement créées.
- Le groupe Script : La commande Valider le script valide la syntaxe du script saisi.

Eléments de script disponibles contient tous les articles de script disponibles pour bâtir le script. Double cliquer sur l'élément ou cliquer sur le bouton Insérer pour insérer l'élément à l'endroit où le curseur se trouve dans le script.

Description de l'élément donne une information basique sur l'élément de script inséré.

La liste d'erreurs comporte les erreurs rapportées après utilisation de la commande Valider le script.

5.15.3. Message

Utiliser l'action Message pour écrire des chaînes personnalisées (par exemple, des messages d'avertissement personnalisés, des valeurs de variables et des commentaires). L'action Message crée des saisies personnalisées dans vos fichiers de journal Automation Manager. Les fichiers de journal Automation contiennent les informations, les avertissements et les descriptions d'erreurs générés par l'application. Utiliser les journaux Message pour faire un suivi de vos variables de messages en cours de configuration, de dépannage et de débogage.

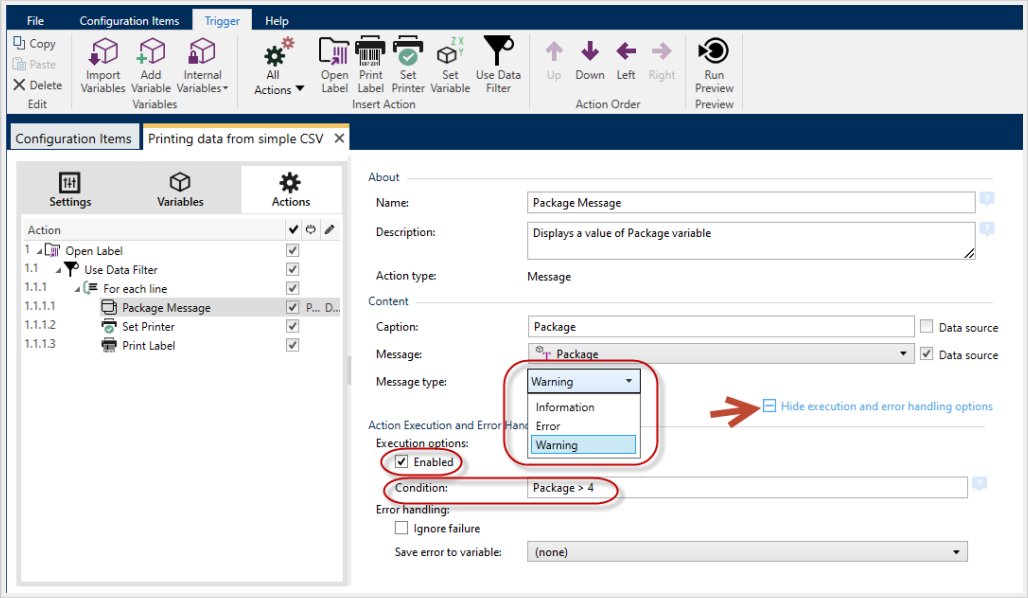
Pour configurer les actions Message, procéder comme suit :

1. Aller dans **Toutes les actions** et sélectionner **Message** dans le menu déroulant **Action**.
2. Renommer votre action et insérer votre description.
3. Configurer le **contenu** de votre message : **Légende**, **Message** et **Type de message**.
   Les **types de messages** comprennent :
    - **Information**
    - **Erreur**
    - **Avertissement**
4. Agrandir **Afficher l'exécution et les options de traitement d'erreur** afin de définir les conditions pour afficher les messages, ignorer les défaillances et enregistrer les erreurs d'Automation en variables.

NiceLabel Automation Manager affiche des messages colorés (par exemple, les erreurs en rouge et les avertissements en orange) dans votre panneau Journal d'Automation.

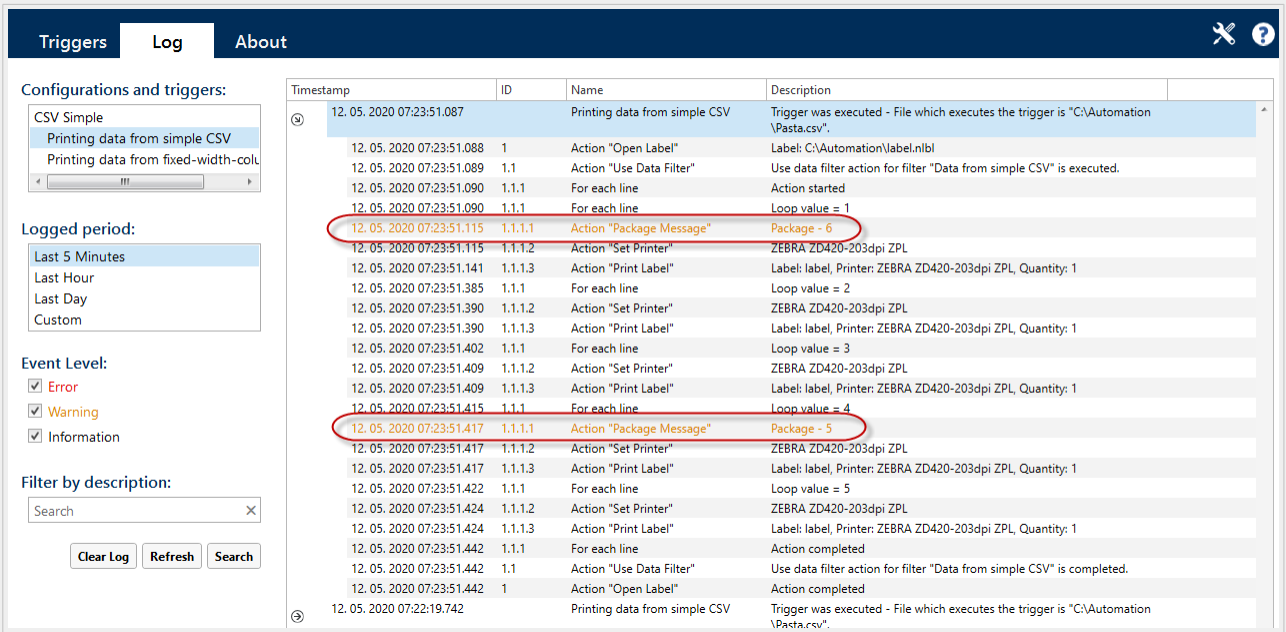
Exemple :

Vous imprimez des étiquettes pour pâtes avec Automation. Votre déclencheur reçoit des valeurs de variables depuis des fichiers CSV générés par le système ERP. Quand votre valeur de variable « Paquet » est supérieure à 4, le journal d'Automation crée un avertissement.



*Configuration des actions **Message**.*

Votre résultat dans Automation Manager se présent comme suit :



NOTE

Quand vous définissez la gravité de Message sur Erreur, vos déclencheurs ne passent pas à l'état d'erreur. Une impression est encore possible.

Utiliser les journaux de Message pour :

- le dépannage de la configuration.
- le débogage de vos solutions
- le suivi des valeurs de vos variables sélectionnées

l'affichage de vos avertissements et messages d'erreur personnalisés

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.15.4. Vérifier la licence

Cette action lit la licence activée et exécute les actions indentées sous cette action seulement si un certain type de licence est utilisé.

> **ASTUCE**
> Cette action protège la configuration des déclencheurs, pour qu'ils ne soient pas utilisés sur des machines non autorisées.

> **NOTE**
> La clé de licence qui active le logiciel peut également encoder l'ID d'une solution. C'est un nombre unique qui identifie le fournisseur de solution qui a vendu la licence NiceLabel 10.

Si l'ID de la solution configurée correspond à l'ID de solution encodée dans la licence, la machine de destination sera autorisée à utiliser les actions indentées, limitant l'exécution aux licences vendues par le fournisseur de la solution.

Les déclencheurs peuvent être cryptés et verrouillés pour que seuls les utilisateurs autorisés puissent ouvrir la configuration. Pour plus d'informations, consulter l'article Protection de la configuration des déclencheurs dans le guide utilisateur de NiceLabel Automation.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Information sur la licence** permet de sélectionner l'ID de la licence.

- **ID de la licence :** Définit le numéro ID des licences qui sont autorisées à exécuter les actions indentées.
  - Si la valeur introduite n'est pas l'ID de la licence qui est encodée dans la licence, les actions indentées ne seront pas exécutées.
  - Si la valeur introduite est 0, les actions seront exécutées quelle que soit la licence valide trouvée.

> **NOTE**
> L'identifiant (UID) de partenaire numérique peut aussi servir d'ID de licence. Cette option n'est disponible que pour les membres du programme de partenariat numérique de NiceLabel.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.15.5. Essayer

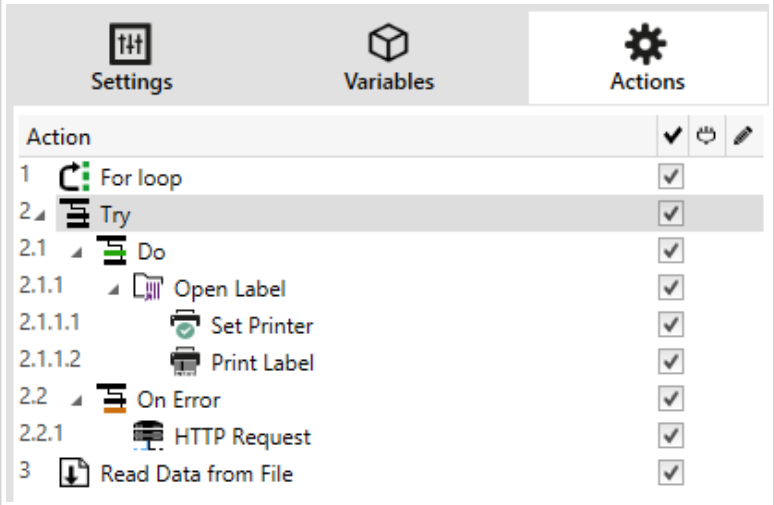
> **INFO NIVEAU DE PRODUIT**
> Les fonctions Automation Builder nécessitent d'utiliser **LMS Enterprise**.

Cette action vous permet de :

- Gérer facilement les erreurs pendant que les actions s'exécutent.
- Lancer un autre ensemble d'actions si une erreur survient.

L'action Essayer comporte deux sous-groupes d'actions, **Faire** et **En cas d'erreur**. Toutes les actions qui doivent s'effectuer quand le déclencheur est activé doivent être placées dans le sous-groupe **Faire**. Si aucune erreur n'est détectée, seules les actions du sous-groupe **Faire** sont exécutées. Toutefois, quand une erreur survient, l'exécution des actions du sous-groupe **Faire** s'interrompt et l'exécution passe aux actions du sous-groupe **En cas d'erreur**.

> **NOTE**
> Vous devez activer Impression synchrone pour trouver les erreurs avec **En cas d'erreur**.



Exemple

Si une des actions du sous-groupe Faire échoue, l'exécution de l'action s'arrête et redémarre avec les actions du sous-groupe En cas d'erreur. Quand Essayer est placé séparément, il peut mettre fin à l'exécution du déclencheur. Dans ce cas, Essayer est indenté sous l'action Boucler. Normalement, toute erreur dans le sous-groupe Faire arrête aussi l'exécution de l'action Boucler, même si d'autres phases de Boucler doivent encore être exécutées. Dans ce cas, l'action Enregistrer les Données dans un Fichier ne sera pas exécutée non plus. Par défaut, toute erreur interrompt le processus complet du déclencheur.

Toutefois, l'exécution de l'itération suivante de l'action Boucler peut continuer. Pour cela, il faut activer Ignorer l'échec dans l'action Essayer. Si les données de l'étape Boucler causent une erreur dans Faire, les actions de En Cas d'erreur s'exécutent. Après ça, l'action Enregistrer les Données dans un Fichier au niveau 2 s'exécute, puis l'action Boucler reprend jusqu'à la prochaine itération.

ASTUCE

Cette action permet de détecter facilement les erreurs et d'exécuter les actions Renvoi d'informations ou Rapports. Par exemple, si une erreur survient durant le traitement du déclencheur, un avertissement peut être envoyé. Pour plus d'informations, consulter l'article Retour de l'état du travail d'impression dans le guide utilisateur de NiceLabel Automation.

NOTE

Important! L'action **Essayer** donne les résultats attendus avec des actions asynchrones. Si votre boucle Essayer inclut l'action Imprimer l'étiquette qui échoue, l'exécution de l'action effectue quand même la boucle Essayer et ne bascule pas sur les actions **En cas d'erreur** comme attendu. La conséquence du non-basculement sur les actions **En cas d'erreur** est l'action Imprimer l'étiquette exécutée en mode synchrone par défaut. Afin d'éviter ça, assurez-vous d'avoir activé l'impression supervisée. Aller dans les paramètres des déclencheurs > **Autre** > **Commentaires du moteur d'impression** puis activer **Impression supervisée**.

Plus d'informations sur l'impression supervisée dans la section Mode d'impression synchrone.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.15.6. Transformation XML

> **INFO NIVEAU DE PRODUIT**
> La fonctionnalité de cet élément est disponible dans LMS Enterprise.

Cette action transforme le document XML en un autre document en utilisant les règles de transformation fournies. Les règles doivent être fournies par la définition .XSLT dans un fichier, ou par une autre source variable.

L'action permet de convertir les documents XML complexes en documents XML ayant une structure plus gérable. XSLT signifie Transformations XSL. XSL signifie Extensible Stylesheet Language, c'est le langage de la feuille de style des documents XML.

L'action Transformer l'XML va stocker le document XML converti dans la variable sélectionnée. Le fichier original reste intact sur le disque. Pour enregistrer le document XML converti, utiliser l'action Enregistrer les données dans un fichier.

NOTE

Utiliser cette action pour simplifier les documents XML fournis par l'application hôte. Définir des filtres XML pour les document XML complexes peut prendre un certain temps, dans certains cas le XML est trop complexe pour être traité. Comme alternative, il faut définir les règles pour convertir XML en une structure facile à traiter par le filtre XML, ou même éliminer complètement la nécessité d'un filtre. Vous pouvez convertir un document XML en un XML supporté de façon native, tel que XML Oracle, et ensuite l'exécuter simplement avec l'action Exécuter le fichier de commande Oracle XML.

ASTUCE

Un exemple de cette action est installé avec le produit. Pour l'ouvrir, aller sur **Aide** > **Fichiers d'exemples** > **Transformer l'XML** et lancer la configuration de Transformations XML.misx. Les détails se trouvent dans le fichier **Readme**.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Source de données** définit les données XML à transformer.

- **Utiliser les données reçues par le déclencheur :** Définit l'utilisation des données reçues par le déclencheur. Le même résultat peut être atteint en activant la variable interne **`DataFileName`** et en utilisant le contenu du fichier auquel elle se réfère. Pour plus d'informations, consulter l'article Utilisation de Valeurs Composées dans le guide utilisateur de NiceLabel Automation.
- **Nom de fichier :** Définit le chemin et nom du dossier contenant le fichier XML à transformer. Le contenu de ce fichier est à utiliser. L'option Source de données permet de définir le nom du fichier dynamiquement. Sélectionner une variable qui contient le chemin et/ou le nom du fichier. Cette action va ouvrir le fichier spécifié et appliquer la transformation au contenu du fichier, qui doit être au format XML.
- **Variable :** Définit la variable (nouvelle ou existante) qui contient le flux d'impression. Le contenu de la variable sélectionnée est utilisé et doit contenir une structure XML.

Le groupe **Source de données des règles de transformation (XSLT)** définit les règles de transformation (document .XSLT) qui seront appliquées au document XML.

- **Nom du fichier :** Définit le chemin et nom du dossier contenant les règles de transformation (.XSLT).
- **Personnalisé :** Définit le contenu personnalisé. Elle peut être fixe, un mixte de contenu fixe et variable, ou contenu variable seul. Pour insérer le contenu d'une variable, cliquer sur le bouton flèche à droite

de la zone de données et insérer une variable de la liste. Pour plus d'informations, consulter l'article Utilisation de Valeurs Composées dans le guide utilisateur de NiceLabel Automation.

Le groupe **Enregistrer le résultat dans la variable** définit la variable dans laquelle sera enregistré le fichier transformé.

- **Variable :** Spécifie la variable qui contiendra le résultat du processus de transformation. Par exemple : si des règles convertissent un XML complexe en un XML plus simple, le contenu de la variable sélectionnée est un fichier en XML simple.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (**`vrai`** ou **`faux`**). Quand le résultat de l'expression est **`vrai`**, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes **`ActionLastErrorId`** et **`ActionLastErrorDesc`**.

5.15.7. Grouper

Utiliser cette action pour regrouper les actions dans un même ensemble. Toutes les actions regroupées dans une action **Grouper** appartiennent au même groupe et s'exécutent ensemble.

Cette action comporte les avantages suivants :

- **Une meilleure organisation et un affichage du flux de travail de l'action.** Chaque action Grouper peut être développée ou réduite pour afficher les actions indentées uniquement en cas de besoin. Cela permet de conserver l'espace de configuration plus propre.
- **Définition d'une exécution sous condition.** Il suffit de définir une condition pour l'action Grouper, plutôt que pour chaque action du groupe. Quand la condition est réalisée, toute les actions du Groupe s'exécutent. Cela permet de gagner du temps pour la configuration et de réduire les erreurs. L'action Grouper est une bonne méthode pour définir l'exécution de IF..THEN (Si..Alors) pour plusieurs actions.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes **`ActionLastErrorId`** et **`ActionLastErrorDesc`**.

5.15.8. Consigner les événements

INFO NIVEAU DE PRODUIT

Les fonctions Automation Builder nécessitent **NiceLabel LMS Enterprise** ou **NiceLabel LMS Pro**.

Cette action consigne les événements dans un journal de NiceLabel Control Center pour en avoir l'historique ou en cas de problème.

NOTE

Pour que cette action soit active, vérifier que la journalisation du travail d'impression dans NiceLabel Control Center est activée.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Données de l'événement** contient les informations sur l'événement à consigner.

- **Information :** Description basique de l'événement qui sera incluse dans le journal des événements de NiceLabel Control Center. Elle peut avoir jusqu'à 255 caractères.
- **Détails :** description détaillée de l'événement à consigner dans le journal de NiceLabel Control Center. Elle peut avoir jusqu'à 2000 caractères.

> **ASTUCE**
> La description saisie dans les champs **Information** et **Détails** permet de filtrer les événements dans **l'historique de toutes les activités** de Control Center. Dans Control Center, aller sur **Historique** > **Toutes les activités** > **Définir un filtre**. Pour plus d'informations, consulter le Guide d'utilisation du Control Center .

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.15.9. Aperçu de l'étiquette

INFO NIVEAU DE PRODUIT

Les fonctions Automation Builder nécessitent d'utiliser **LMS Enterprise**.

Cette action exécute le processus d'impression et prévisualise l'étiquette. Par défaut, l'aperçu est enregistré sur le disque au format JPEG, mais tout autre type d'image est utilisable. Vous pouvez également contrôler la taille de l'aperçu de l'image créée. L'action va générer l'aperçu pour une étiquette.

Une fois créé l'aperçu de l'étiquette dans un fichier, ce fichier peut être envoyé à une application tierce en utilisant une des actions de sortie, telle que Envoyer les données au HTTP, Envoyer les données au port série, Envoyer les données vers un port TCP/IP ou utilisé comme message de réponse pour les déclencheurs bidirectionnels, tels que Déclencheur de serveur HTTP et déclencheur Web Service. L'application tierce peut prendre l'image et la montrer à l'utilisateur comme aperçu de l'étiquette.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Aperçu** définit le fichier de l'étiquette à prévisualiser.avec les détails.

- **Nom du fichier :** Spécifie le chemin et nom du fichier. S'il est codé en dur, le même fichier sera utilisé à chaque fois. En utilisant le nom de fichier sans le chemin, le dossier où est sauvegardé le fichier de configuration (.MISX) est utilisé. En utilisant une référence relative au nom de fichier, le dossier avec le fichier .MISX est utilisé comme dossier racine. **Source de données** active le nom de fichier variable. Sélectionner une variable qui contient le chemin et/ou le nom du fichier quand le déclencheur est exécuté. En général, la valeur est assignée à la variable par un filtre.
- **Type d'image :** spécifie le type d'image utilisé pour enregistrer l'aperçu de l'étiquette.
- **Aperçu du verso de l'étiquette (étiquettes double-face) :** permet de prévisualiser le verso de l'étiquette. C'est utile pour avoir l'aperçu du verso dans les étiquettes recto-verso.

Exemple

Par exemple, si le masque de l'étiquette définit des dimensions de 4" × 3" et l'imprimante d'étiquettes a une résolution de 200 DPI, l'aperçu d'image résultant aura les dimensions de 800 × 600 pixels. La largeur égale à 4 pouces multipliée par 200 DPI, donne un résultat de 800 pixels. La hauteur égale à 3 pouces multipliée par 200 DPI, donne un résultat de 600 pixels.

Le groupe **Paramètres additionnels** permet d'utiliser des valeurs provisoires.

- **Utiliser des valeurs provisoires :** remplace les valeurs des données manquantes par des valeurs provisoires et les affiche dans l'aperçu.

> **ASTUCE**
> **Valeur provisoire** définit une valeur variable dans un espace personnalisé d'un objet lors de la création des étiquettes ou formulaires. Dans un objet de l'étiquette, la valeur provisoire est remplacée par la valeur réelle de la variable au moment de l'impression.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

> **NOTE**
> Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

5.15.10. Créer une variante d'étiquette

INFO NIVEAU DE PRODUIT

Les fonctions Automation Builder nécessitent d'utiliser **LMS Enterprise**.

Cette action permet de créer un double de l'étiquette existante avec les données en clair. Dans les objets de l'étiquette, les valeurs des sources de données sont verrouillées. Ces valeurs sont définies par la source de données applicable.

L'objectif de cette étiquette avec les données « verrouillées » en clair est d'avoir une étiquette adaptée aux processus d'approbation ayant besoin d'approuver simultanément les données et le masque. Au lieu d'afficher une étiquette sans contenu défini pour les objets, l'approbateur examine une variante comportant les valeurs définies. Cela lui permet de voir rapidement et d'approuver la mise en page finale de l'étiquette avec les valeurs qui vont être utilisées pour l'impression.

ASTUCE

Le processus d'approbation des étiquettes s'applique aux étiquettes stockées dans le Stockage de documents du Control Center. Différents types de flux d'approbation peuvent s'appliquer aux étiquettes stockées et à leurs variantes. Le choix du flux d'approbation dépend des contraintes liées à l'activité. Pour plus de détails, consulter le guide utilisateur NiceLabel 10 du Control Center.

À propos : Ce groupe identifie l'action sélectionnée.

- **Nom** : Permet de définir un nom pour l'action. Cela permet de reconnaître plus facilement l'action dans la liste des actions de la solution. Par défaut, le nom de l'action provient de son type.
- **Description** : Informations personnelles sur l'action. Mettre une description qui explique l'objectif et le rôle de l'action dans la solution.
- **Type d'action** : Information en lecture seule sur le type d'action sélectionné.

Le groupe **Paramètres** définit le fichier de l'étiquette à convertir et le fichier de sortie (variante).

- **Nom de l'étiquette :** Le nom du fichier d'étiquette à convertir en variante lisible avec les valeurs des sources de données verrouillées. La **Source de données** définit dynamiquement le **Nom de l'étiquette** en utilisant une variable nouvelle ou existante.
- **Sources de données pour l'impression:** Cette option permet de définir les sources de données fournissant les valeurs au moment de l'impression. Quand une source de données est listée dans ce champ, sa valeur n'est pas verrouillée. Elle peut être fournie au moment de l'impression. Exemple : les sources de données de production comme le lot, la date d'expiration, etc.

  ASTUCE

  Mettre uniquement les noms des sources de données, sans crochet, séparées par des virgules ou mis en colonne avec la touche Entrée.

- **Nom du fichier de sortie :** Nom du fichier de la variante de l'étiquette qui permettra la lecture. La **Source de données** définit dynamiquement le **Nom de l'étiquette** en utilisant une variable nouvelle ou existante.

Plusieurs règles s'appliquent à la variante de l'étiquette :

1. Par défaut, les valeurs de source de données sont verrouillées. Pour éviter qu'elles soient verrouillées, il faut les lister dans le champ **Sources de données à l'impression** pour qu'elles restent actives sur la variante de l'étiquette. Leurs valeurs pourront être définies au moment de l'impression.
2. Les variables compteurs, les fonctions, les champs de base de données et les variables globales sont convertis en variables non saisies.
3. Les graphiques sont intégrés.
4. La variante de l'étiquette de destination placée dans le Stockage de documents NiceLabel Control Center est automatiquement activée. Le **Nom de l'étiquette** originale et les **Sources de données à l'impression** sont utilisés en commentaire pour vérification.
5. Les variantes d'étiquettes peuvent être ouvertes dans NiceLabel 10 Designer, mais elles sont verrouillées.
6. Les fichiers d'étiquettes générés avec cette action ne peuvent pas être importés.
7. Quand une variante d'étiquette est stockée dans la mémoire de l'imprimante, la commande de rappel donne seulement la valeur des sources de données saisies à l'impression.
8. En utilisant NiceLabel Control Center, l'aperçu de l'étiquette dans le Stockage de documents permet de modifier les sources de données saisies à l'impression.
9. Les variables de temps actuel et date du jour ne peuvent être définies comme sources de données saisies à l'impression sur la variante de l'étiquette.

Exécution d'une action et traitement d'erreur

Chaque action peut être soumise à condition. Une action conditionnelle ne fonctionne que quand les conditions fournies lui permettent de fonctionner. Pour définir ces conditions, cliquer sur **Afficher les options d'exécution et de gestion des erreurs**.

Options d'exécution existantes :

- **Activée :** Spécifie si l'action est activée ou désactivée. Seules les actions activées seront exécutées. Cette fonctionnalité est utilisable lors du test d'un formulaire.
- **Condition** : Définit une expression dont le résultat est une valeur bouléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. Une condition permet d'éviter que l'action soit exécutée chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Spécifie si une erreur doit être ignorée ou non. Quand **Ignorer l'échec** est activé, l'exécution des actions continue même si l'action en cours a échoué.

NOTE

Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

Exemple

À la fin de l'impression, vous pouvez envoyer la mise à jour du résultat à une application externe en utilisant l'action **Requête HTTP**. Si l'action d'impression échoue, le déclencheur arrête le traitement des actions. Pour effectuer le rapport, même après un échec d'impression, l'action **Imprimer l'étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Enregistrer l'erreur dans une variable** : permet de définir une variable pour y enregistrer l'erreur. La raison de l'erreur est aussi enregistrée dans les variables internes **`ActionLastErrorId`** et **`ActionLastErrorDesc`**.

6. Exécuter et gérer les déclencheurs

6.1. Déployer la configuration

Après avoir configuré et testé les déclencheurs dans Automation Builder, il faut déployer la configuration dans le service NiceLabel Automation et démarrer les déclencheurs. À ce moment, les déclencheurs sont activés et commencent à surveiller les événements définis.

Pour déployer la configuration, utiliser les méthodes suivantes.

Déployer à partir de Automation Builder

1. Démarrer Automation Builder.
2. Charger la configuration.
3. Aller à l'onglet **Éléments de configuration**.
4. Cliquer sur le bouton **Déployer la configuration** dans le groupe du ruban Déployer.
   La configuration se chargera dans Automation Manager fonctionnant sur la même machine.
5. Démarrer les déclencheurs que vous voulez activer.

Si cette configuration a déjà été chargée, le déploiement force le rechargement, tandis que les déclencheurs restent actifs.

Déployer à partir de Automation Manager

1. Démarrer Automation Manager.
2. Aller à l'onglet **Déclencheurs**.
3. Cliquer sur le bouton **+Ajouter** et rechercher la configuration sur le disque.
4. Démarrer les déclencheurs qu'il faut activer.

Déployer depuis la ligne de commande

Pour déployer la configuration **`C:\Project\Configuration.MISX`** et exécuter le déclencheur nommé **`CSVTrigger`**, effectuer les opérations suivantes :

```
NiceLabelAutomationManager.exe ADD c:\Project\Configuration.MISX
NiceLabelAutomationManager.exe START c:\Project\Configuration.MISX CSVTrigger
```

Pour plus d'informations, consulter l'article Contrôler le Service avec les paramètres de ligne de commande.

6.2. Options de journalisation des événements

INFO NIVEAU DE PRODUIT

Certaines des fonctionnalités décrites dans cet article nécessitent l'achat de produits **NiceLabel LMS**.

NiceLabel Automation enregistre les événements à divers endroits, selon le scénario de déploiement. Les deux premiers journaux sont disponibles pour tous les niveaux de produits NiceLabel Automation.

- **Journaliser dans la base de données du journal :** La journalisation dans la base de données du journal interne est toujours activée. La base de données du journal interne garde une trace détaillée de tous les événements. Pour visualiser les informations journalisées, utiliser des filtres pour afficher les événements correspondants correspondant aux règles. Pour plus d'informations, consulter l'article Utilisation du journal d'événements.
  Les données sont sauvegardées dans la base de données SQLite. C'est un journal référentiel temporaire, les événements sont supprimés de la base de données une fois par semaine. La période de nettoyage est configurable dans les Options. Les enregistrements des anciens événements sont effacés de la base de données, mais la base de données n'est pas compactée (vidée), ils peuvent donc encore occuper de la place sur le disque. Pour la compacter, utiliser un logiciel tiers de gestion SQLite.
- **Journaliser dans le journal d'événements de l'Application Windows :** Les événements importants sont enregistrés dans le journal d'événements de l'Application Windows au cas où NiceLabel Automation n'arrive pas à démarrer. Vous disposez ainsi d'une autre ressource pour les événements journalisés.
- **Journaliser vers Control Center :** Le journal du Control Center se trouve dans les produits **LMS Enterprise** et **LMS Pro** . Control Center est une console de gestion Web qui enregistre tous les événements sur un ou plusieurs serveurs NiceLabel Automation. Les données sont sauvegardées dans la base de données du Serveur SQL de Microsoft. Vous pouvez aussi rechercher dans les données collectées, et il dispose aussi d'alertes automatisées pour certains événements, d'une gestion des imprimantes, d'un stockage de documents, d'un système de contrôle des versions, de flux de travail et de la réimpression d'étiquettes.

NOTE

Pour plus d'informations, consulter le Control Center guide utilisateur.

6.3. Gestion des déclencheurs

Automation Manager est la partie de gestion du logiciel NiceLabel Automation. En utilisant Automation Builder pour configurer les déclencheurs, vous utilisez Automation Manager pour les déployer et les exécuter dans l'environnement de production. L'application permet de charger des déclencheurs de différentes configurations, de voir leur état en direct, de les démarrer/arrêter et de voir les détails de leur exécution dans le fichier du journal.

Il est possible de modifier l'affichage des configurations chargées et leurs déclencheurs. Le dernier affichage est mémorisé, il s'appliquera au prochain lancement de Automation Manager. Si vous activez l'affichage **Par état**, les déclencheurs de toutes les configurations ouvertes qui partagent le même l'état s'affichent ensemble. Si vous activez l'affichage par **Configuration**, les déclencheurs de la configuration sélectionnée s'affichent en même temps, sans tenir compte de leur état. L'état du déclencheur est affiché en couleur dans l'icône du déclencheur, permettant une identification plus aisée.

Les détails du déclencheur affiché changent en temps réel, à la détection des événements du déclencheur. Parmi les informations qui s'affichent, il y a le nom du déclencheur, le type de déclencheur, combien d'événements ont déjà été exécutés, combien d'erreurs ont été détectées et le temps écoulé depuis le dernier événement. En passant la souris sur le nombre de déclencheurs déjà exécutés, le nombre d'événements du déclencheur en attente d'exécution devient visible.

NOTE

La configuration chargée est en mémoire cache. Quand la configuration est modifiée dans Automation Builder, Automation Manager ne l'applique pas automatiquement. Pour appliquer le changement, recharger la configuration.

Charger la configuration

Pour charger la configuration, cliquer sur le bouton **+Ajouter** et rechercher le fichier de configuration (.MISX). Les déclencheurs de la configuration sont chargés en état suspendu. Il faut alors démarrer les déclencheurs pour les activer. Pour plus d'informations, voir la section Déployer la configuration.

La liste de configurations chargées et l'état de chaque déclencheur est mémorisé. Si le serveur est redémarré pour une raison quelconque, le Service NiceLabel Automation rétablit l'état que le déclencheur avait avant le démarrage.

Rechargement et enlèvement de la configuration

Après avoir mis à jour et enregistré la configuration dans Automation Builder, les changements ne s'appliquent pas automatiquement dans Automation Manager. Pour recharger la configuration, cliquer à droite sur le nom de la configuration, puis sélectionner **Recharger la configuration**. Cette action recharge tous les déclencheurs. Si la mise en cache des fichiers est activée, le rechargement force la synchronisation de tous les fichiers utilisés par les déclencheurs.

Démarrer/arrêter les déclencheurs

Si vous chargez des déclencheurs à partir d'une configuration, leur état par défaut est arrêté. Pour démarrer le déclencheur, cliquer sur le bouton **Démarrer** dans la zone des déclencheurs. Pour arrêter le déclencheur, cliquer sur **Arrêter**. Plusieurs déclencheurs d'une même configuration peuvent être démarrés/arrêtés simultanément.

Une ligne de commande d'une configuration peut aussi contrôler le démarrage/arrêt. Pour plus d'informations, consulter l'article Contrôler le Service avec les paramètres de ligne de commande.

Gestion des conflits de déclencheurs

Les déclencheurs peuvent être en état d'erreur en raison de situations listées ci-dessous. Un déclencheur en erreur ne peut pas démarrer avant d'avoir résolu le problème.

- **Déclencheur non configuré correctement ou entièrement :** Dans ce cas, le déclencheur n'est pas configuré, les caractéristiques obligatoires ne sont pas définies, ou les actions définies pour cette imprimante ne sont pas configurées. Il est impossible de démarrer un tel déclencheur.
- **La configuration du déclencheur se croise avec un autre déclencheur :** Deux déclencheurs ne peuvent pas surveiller le même événement.

Exemple

Deux déclencheurs de fichier ne peuvent pas surveiller le même fichier. Deux déclencheurs HTTP ne peuvent pas accepter de données sur le même port. Si la configuration du déclencheur se croise avec un autre déclencheur, le second déclencheur ne fonctionne pas, car l'événement est déjà capturé par le premier déclencheur. pour plus d'informations, consulter le panneau du Journal de ce déclencheur.

Réinitialisation de l'état d'erreur

Quand l'exécution du déclencheur cause une erreur, la couleur de l'icône du déclencheur devient rouge, le déclencheur est en état d'erreur, et les détails de l'événement sont enregistrés dans le journal de la base de données. Même si tous les événements suivants se terminent avec succès, le déclencheur reste en état d'erreur jusqu'à ce que vous confirmiez la saisie de l'erreur et la modification de l'état. Pour confirmer l'erreur, cliquer sur l'icône à côté du compteur d'erreurs sous les détails du déclencheur.

Utilisation du panneau de notifications

Le panneau de notifications est la zone où les messages importants s'affichent. Il est situé au-dessus de la liste de déclencheurs dans l'onglet Déclencheurs. La zone de notifications affiche les **messages d'état**, tels que « Mode d'essai » ou « Mode d'essai expiré », ou les **messages d'avertissement** tels que « Le traçage a été activé ».

Visualisation des données du journal

Chaque activité du déclencheur est enregistrée dans la base de données du journal, y compris les événements démarrage/arrêt du déclencheur, l'exécution avec succès des actions et les erreurs rencontrées durant l'exécution. Cliquer sur le bouton Journal pour visualiser les événements enregistrés pour le déclencheur sélectionné. Pour plus d'informations, consulter l'article Utilisation du journal d'événements.

6.4. Utilisation du journal d'événements

Toutes les activités du logiciel NiceLabel Automation sont enregistrées dans la base de données du journal pour activer l'historique et le dépannage. En cliquant sur le bouton **Journal** dans l'onglet Déclencheurs, les événements de ce déclencheur s'affichent. Le journal affiche les informations concernant tous les événements liés au filtre défini.

Les données du journal sont utiles pour la résolution de problèmes. Si une action du déclencheur ne peut pas être exécutée, l'application enregistre une description de l'erreur dans le fichier du journal, ce qui permet d'identifier et résoudre le problème.

NOTE

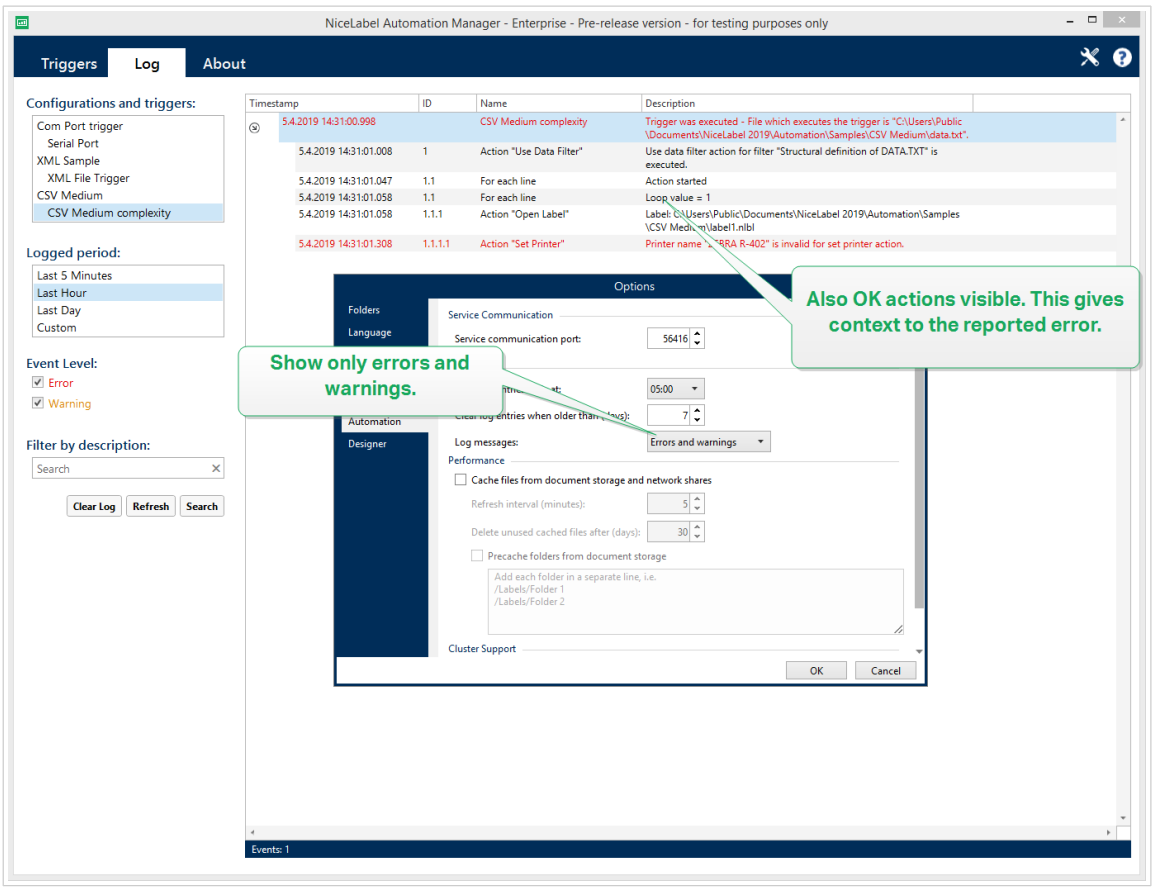
La rétention de données par défaut est de 7 jours et est configurable dans les Options. Pour réduire la taille de la base de données du journal sur les systèmes chargés, il suffit de diminuer la période de rétention.

Filtrage des événements

Les filtres configurables :

- **Configuration et déclencheurs :** Spécifie quels événements afficher, les événements du déclencheur sélectionné ou les événements de tous les déclencheurs de la configuration sélectionnée.
- **Période enregistrée :** Spécifie la période durant laquelle les événements se sont déroulés. Par défaut, cette période correspond aux **5 dernières minutes**.
- **Niveau des événements :** Spécifie le type (l'importance) des événements à afficher :
  - **Erreur** est le type d'événement qui interrompt l'exécution.
  - **Avertissement** est le type d'événement dans lequel des erreurs surviennent, mais elles sont configurées pour être ignorées.
  - **Informations** est le type d'événement qui enregistre toutes les informations non erronées.

> **NOTE**
> En cas d'erreurs et d'avertissements, Automation affiche également la séquence complète d'actions exécutées dans un déclencheur.



Le niveau du journal est configurable dans les **Options**.

- **Filtrer par description :** Vous pouvez afficher tous les événements qui contiennent la chaîne de caractères donnée. Utiliser cette option pour la résolution d'erreurs sur des déclencheurs chargés. Le filtre s'applique au champ de description du déclencheur.

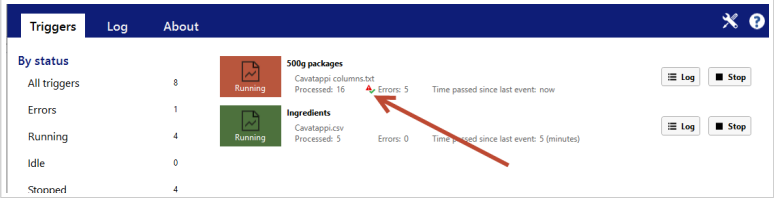
Effacement de la base de données du journal

Effacer le journal dans Automation Builder. Pour effacer la base de données du journal, cliquer sur le bouton **Effacer le journal**.

> **AVERTISSEMENT**
> Utiliser ce bouton avec précaution, c'est une action irréversible. Elle enlèvera **TOUS** les événements enregistrés dans la base de données du journal, et s'appliquera à tous les déclencheurs, pas seulement au déclencheur en cours.

Nettoyages automatisés de la base de données du journal

Automation vous permet de définir des nettoyages automatisés réguliers pour les entrées du journal à propos des déclencheurs exécutés avec succès. L'agrandissement de la base de données du journal réduit à coup sûr les performances du système.



La réinitialisation du compteur ne supprime pas les erreurs du journal.

Nettoyages automatisés de la base de données des journaux

Automation vous permet de configurer des nettoyages automatisés réguliers pour les entrées de journal concernant les déclencheurs exécutés avec succès. C'est ainsi que vous vous assurez que la croissance de la base de données des journaux réduit les performances du système.

Pour planifier les nettoyages automatisés de la base de données des journaux :

1. Fichier ouvert **`produit.config`** dans l'éditeur de texte.
   Le fichier se trouve ici :

   ```
   %PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
   ```

2. Créer une copie de sauvegarde du **`produit.config`** fichier.

3. Automation utilise deux paramètres pour nettoyer les déclencheurs. Ajoutez ces deux paramètres à votre fichier product.config.

   - **`/IntegrationService/LogSuccessfulTriggerPurgeInterval>`**. Ce paramètre définit la longueur de l'intervalle de temps entre deux nettoyages consécutifs. Tapez la durée de l'intervalle en minutes.

   - **`/IntegrationService/LogSuccessfulTriggerPurgeRemovalAge>`**. Ce paramètre vérifie l'âge des messages sur les actions exécutées avec succès.

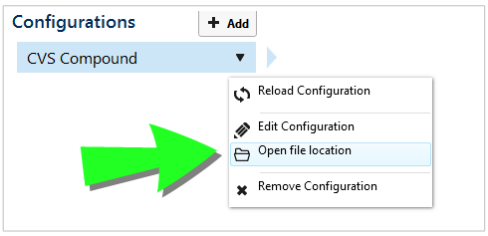
```
<configuration>
    <IntegrationService>
        <LogSuccessfulTriggerPurgeInterval>1</
LogSuccessfulTriggerPurgeInterval>

<LogSuccessfulTriggerPurgeRemovalAge>1<LogSuccessfulTriggerPurgeRemovalAg
e>
    </IntegrationService>
</configuration>
```

6.5. Si votre configuration ne parvient pas à se charger...

Déployée, la configuration de votre Automation est exécutée en tant que processus Windows en arrière-plan. L'Automation Manager avec lequel vous gérez et surveillez votre configuration n'est qu'une interface qui représente les services d'Automation. Dans certains cas, la configuration que vous avez développée, testée et déployée ne parvient pas à se charger. Il y a plusieurs raisons possibles. Suivre les solutions proposées pour que votre configuration d'Automation fonctionne correctement :

1. Le fichier de configuration a été retiré, renommé ou déplacé. Vous pouvez vérifier sur quel fichier vous avez déployé les points de la configuration dans Automation Manager :

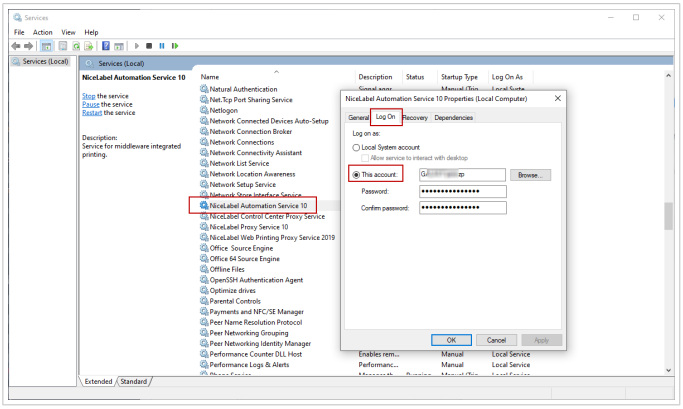


Assurez-vous que le fichier de configuration .misx est disponible à l'emplacement spécifié, et qu'il a gardé le même nom que dans Automation Manager. Si vous avez déplacé ou renommé le fichier de configuration, l'ouvrir dans Automation Builder et redéployer la configuration.

2. Le fichier de configuration se trouve à un emplacement réseau inaccessible en raison d'un problème de connexion au réseau. Vérifier la connectivité au réseau de votre ordinateur/serveur qui stocke la configuration.

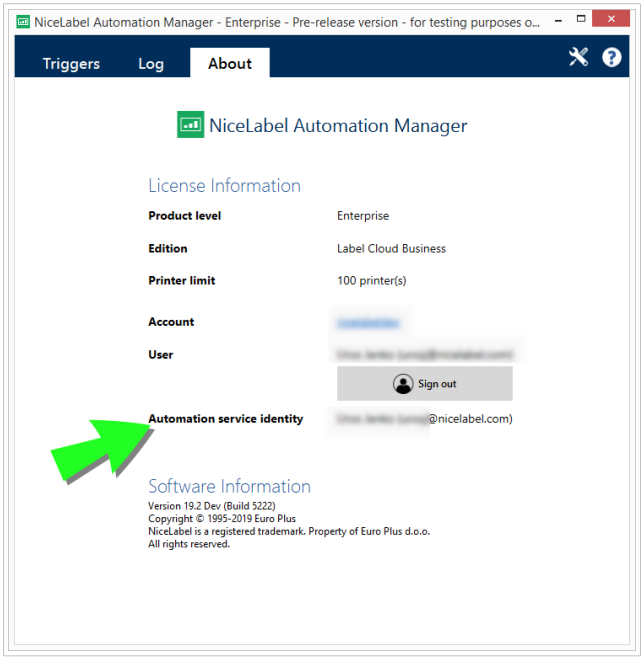
3. **Le service Automation n'a pas le droit d'accès au fichier de configuration.** Vérifier les permissions pour le compte utilisateur utilisé par le service Automation. Cette erreur indique un problème avec le service Automation exécuté en arrière-plan.
   Solutions possibles:

   - Si votre fichier de configuration est stocké sur un disque local ou un partage réseau, vérifier que votre configuration est exécutée avec les identifiants de l'utilisateur ou du domaine local. Exécuter votre configuration sous le Compte système local peut restreindre l'accès aux dossiers et imprimantes partagé(e)s en réseau. Ouvrir Services et voir les propriétés de NiceLabel Automation Service10.

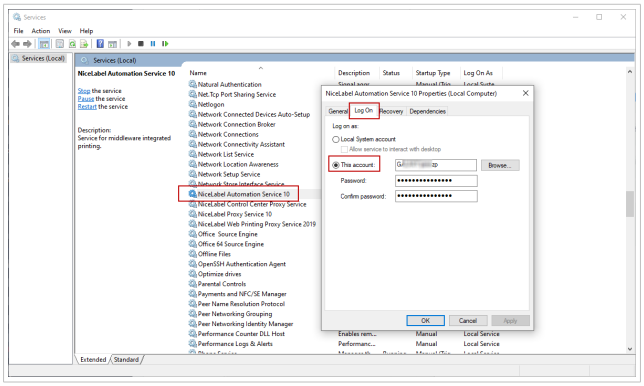


   - Si votre fichier de configuration est stocké dans le Stockage de documents de votre Control Center, il y a plusieurs scénarios possibles :

     - Votre Control Center utilise l'authentification de l'application. La raison pour laquelle votre configuration ne parvient pas à se charger provient d'une erreur d'**identité du service Automation**. L'identité du service Automation doit correspondre à l'identité de l'utilisateur telle que défini dans votre Control Center.
       Vous trouverez votre identité du service Automation dans **Automation Manager** > **onglet À propos**.

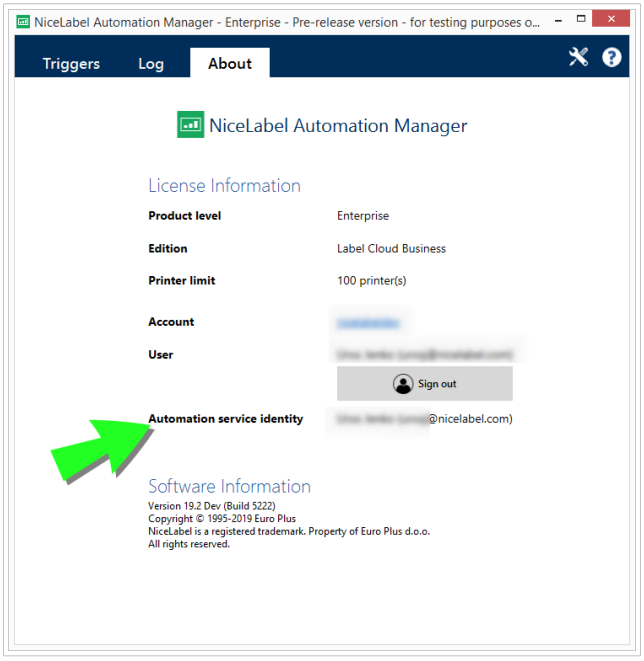


- Votre Control Center utilise l'authentification Windows. Votre configuration ne parvient pas à se charger car vous tentez de l'exécuter en tant qu'utilisateur n'ayant pas les droits suffisants sur votre Control Center. Vérifier l'utilisateur avec lequel vous exécutez le service Automation. Ouvrir **Services** et voir les propriétés de **NiceLabel Automation Service 2019**.



> **NOTE**
> Dans les deux cas, voir le Guide utilisateur NiceLabel Control Center pour plus détails sur les méthodes d'authentification et les droits utilisateur disponibles.

- Votre Control Center est exécuté dans NiceLabel Cloud. La raison pour laquelle votre configuration ne parvient pas à se charger provient d'une erreur d'identité du service Automation. L'identité du service Automation doit correspondre à la connexion à NiceLabel Cloud de l'utilisateur.
  Vous trouverez votre **identité du service Automation** dans **Automation Manager** > **onglet À propos**.


NiceLabel Automation Manager - Enterprise - Pre-release version - for testing purposes o...
Triggers
Log
About
NiceLabel Automation Manager
License Information
Product level
Enterprise
Edition
Label Cloud Business
Printer limit
100 printer(s)
Account
User
Sign out
Automation service identity
@nicelabel.com)
Software Information
Version 19.2 Dev (Build 5222)
Copyright © 1995-2019 Euro Plus
NiceLabel is a registered trademark. Property of Euro Plus d.o.o.
All rights reserved.

7. Performances et options de retour d'informations

7.1. Traitement parallèle

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise**, **LMS Pro**, **Label Cloud Business** et **Label Cloud Compliance**.

NiceLabel Automation supporte le traitement parallèle pour le traitement d'entrée et de sortie. L'efficacité maximum est ainsi garantie pour n'importe quel système avec le logiciel installé. NiceLabel Automation exécute plusieurs tâches simultanément, tout en gardant l'ordre dans lequel les déclencheurs ont été ajoutés. La capacité de traitement du travail d'impression d'étiquettes dépend en grande partie du matériel utilisé.

Traitement parallèle d'entrée

Vous pouvez utiliser plusieurs déclencheurs sur le même appareil. Ils répondent tous simultanément aux changements liés aux événements surveillés. Chaque déclencheur mémorise les données des événements non traités dans la liste d'attente. La liste va mettre les données entrantes en mémoire tampon dans le cas où aucun des travaux d'impression n'est actuellement disponible. Dès qu'un travail d'impression est disponible, il prend le premier flux d'impression qui se trouve dans la file d'attente suivant le principe FIFO (Premier entré, Premier sorti). Ainsi, l'ordre du traitement des données d'entrée est respecté. Mais ça ne garantit pas le principe FIFO pour l'impression. Voir le prochain article ci-dessous.

NOTE

Le traitement parallèle signifie bien plus que d'exécuter plusieurs déclencheurs à la fois. Mais chaque déclencheur peut aussi permettre des connexions concurrentes. Les déclencheurs TCP/IP, HTTP et Web Service acceptent tous les connexions concurrentes de plusieurs clients. De plus, le déclencheur fichier peut être configuré pour surveiller un ensemble de fichiers dans un dossier. La configuration se fait par un masque.

Traitement parallèle de sortie

Généralement, le résultat du déclencheur est le processus d'impression de l'étiquette. Pour ce processus, vous utilisez les données reçues par le déclencheur pour les imprimer sur les étiquettes. Le service NiceLabel Automation exécute les processus d'impression (c.-à-d. « moteurs d'impression ») en parallèle en arrière-plan. Les processeurs modernes ont deux ou plusieurs processeurs centraux d'exécution appelés « cœurs ». Plusieurs cœurs peuvent exécuter plusieurs instructions en même temps, ce qui augmente la vitesse de traitement dans son ensemble. Dans le cas de NiceLabel Automation, plusieurs cœurs augmentent le traitement du travail d'impression, et donc les performances de l'impression d'étiquettes.

Par défaut, chaque instance de NiceLabel Automation exécute chaque processus d'impression en tant que thread séparé sur chaque cœur disponible. Plus le processeur central est puissant, plus la capacité de traitement est élevée. Ceci optimise l'usage de la puissance disponible du processeur central. Le logiciel s'installe avec les paramètres par défaut en définissant que chaque cœur disponible fournisse un seul thread pour le traitement de l'impression. En temps normal, il est inutile de modifier les paramètres par défaut. Si la configuration nécessite d'être modifiée, voir l'article Changer les paramètres d'impression multi threads par défaut.

En cas de grande quantité de processus d'impression, les données du premier événement sont imprimées par un processus d'impression, tandis que les données du second événement sont imprimées simultanément par un autre processus d'impression, si un deuxième processus d'impression est disponible à ce moment-là. Si le second événement ne fournit beaucoup de données, le processus d'impression envoie les données pour l'imprimante plus rapidement que le premier processus d'impression, rompant l'ordre. Dans ce cas, les données du deuxième événement pourraient être imprimées avant les données du premier événement. Pour garantir le principe FIFO pour l'impression, voir l'article Mode d'impression synchrone.

7.2. Mise en cache de fichiers

Pour améliorer le temps de sortie de la première étiquette et les performances générales, NiceLabel Automation permet la mise en cache de fichiers. Il y a souvent des délais d'impression quand les étiquettes, images et bases de données sont chargées depuis un réseau partagé. NiceLabel Automation doit extraire tous les fichiers nécessaires avant de pouvoir démarrer le processus d'impression.

Il y a deux niveaux complémentaires de mise en cache.

- **Mémoire cache :** La mémoire cache stocke les fichiers déjà utilisés. Les étiquettes qui ont été utilisées au moins une fois sont chargées en mémoire cache. Quand le déclencheur requiert l'impression de la même étiquette, l'étiquette est immédiatement disponible pour le processus d'impression. La mémoire cache est activée par défaut. Son contenu est vidé après avoir supprimé ou rechargé une configuration. Les changements d'un fichier d'étiquette sont vérifiés pour chaque action Ouvrir l'étiquette. Si une étiquette plus récente est disponible, elle est automatiquement chargée, remplaçant l'ancienne version en cache.

> **NOTE**
> Une étiquette qui n'est pas utilisée pendant 8 heures est déchargée de la mémoire cache.

- **Cache persistant :** Le cache persistant stocke les données sur le disque. Son rôle est de fournir un stockage de fichiers à moyen terme. Le cache est géré par objet de fichier. Quand un fichier est requis par le partage réseau, le service commence par vérifier si le fichier est déjà présent en cache et l'utilise. Si le fichier n'est pas en cache, il est extrait du partage réseau et mis en cache pour une utilisation ultérieure. Le service de mise en cache met continuellement à jour le contenu du cache avec les versions plus récentes des fichiers. Il est possible de configurer l'intervalle de temps pour la vérification des versions dans le menu des Options.

Prolongement des périodes pour le déchargement des étiquettes

Lorsque l'étiquette est utilisée pour la première fois, elle est chargée dans la mémoire cache. L'étiquette sera disponible pour l'impression instantanée la prochaine fois qu'elle sera requise. Le nettoyage de la mémoire cache supprime toutes les étiquettes non utilisées depuis 8 heures.

Pour prolonger ce temps dans la mémoire cache, effectuer les opérations suivantes :

1. Rechercher le dossier système de NiceLabel Automation.
   `%PROGRAMDATA%\NiceLabel\NiceLabel 10`
2. Effectuer une copie de sauvegarde du fichier **`product.config`**.
3. Ouvrir **`product.config`** dans un éditeur de texte. Le fichier a une structure XML.
4. Ajouter l'élément **`Common/FileUpdater/PurgeAge`**.
5. Ce paramètre définit le nombre de secondes qui définissent la période de stockage des étiquettes dans la mémoire cache. NiceLabel Automation garde une trace du temps écoulé depuis la dernière impression de chaque étiquette. Quand ce laps de temps atteint le seuil défini, l'étiquette est retirée de la mémoire cache.

   > **NOTE**
   > Valeur par défaut : 28800 (8 heures). La valeur maximale est de 2147483647.

   Le fichier **`product.config`** doit avoir le contenu suivant :

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

6. Après la sauvegarde du fichier, le Service NiceLabel Automation applique les paramètres automatiquement.

Activation du cache persistant

> **INFO NIVEAU DE PRODUIT**
> La fonctionnalité de cet élément est disponible dans LMS Enterprise.

Pour activer et configurer le cache persistant, ouvrir l'option, sélectionner NiceLabel Automation et activer **Mettre en cache les fichiers distants**.

- **Mettre à jour les fichiers en cache :** Définit l'intervalle de temps en minutes durant lequel le cache sera synchronisé avec les fichiers dans leur dossier d'origine. Ceci est l'intervalle de temps durant lequel le système peut utiliser l'ancienne version du fichier.
- **Effacer les fichiers en cache lorsqu'ils ont plus de :** Définit l'intervalle de temps en jours pour la suppression de tous les fichiers en cache auxquels vous n'avez pas accédé pendant la durée spécifiée.

NiceLabel Automation utilise le dossier local suivant comme cache pour les fichiers distants :

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10\FileCache
```

> **NOTE**
> On peut mettre en cache des fichiers d'étiquettes ou d'images. Après avoir activé la mise en cache des fichiers, redémarrer le service Automation pour que la modification prenne effet.

Forcer le rechargement du contenu du cache
NiceLabel Automation met à jour automatiquement le contenu du cache après l'intervalle de temps défini. Par défaut, les 5 dernières minutes.

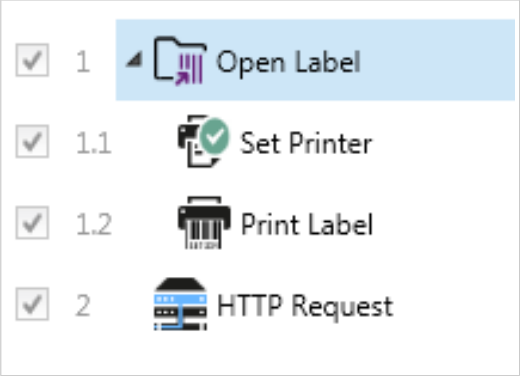
Pour forcer le rechargement du cache manuellement, effectuer les opérations suivantes :

1. Ouvrir Automation Manager.
2. Localiser la configuration qui contient le déclencheur pour lequel il faut forcer le rechargement des étiquettes.
3. Cliquer à droite sur la configuration.
4. Sélectionner **Recharger la Configuration**.

7.3. Traitement d'erreur

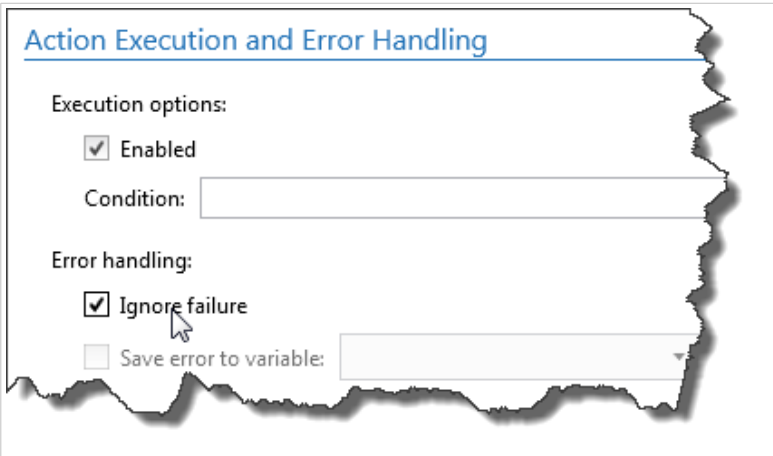
Quand une erreur survient durant l'exécution d'une action, NiceLabel Automation arrête l'exécution de toutes les actions du déclencheur. Si des actions sont définies après que l'action actuelle ait signalé une erreur, ces actions ne sont pas exécutées.

Par exemple, les actions sont définies comme sur la copie d'écran ci-dessous. Si l'action **Définir l'imprimante** échoue, pour une erreur de nom ou une imprimante inaccessible, les actions **Imprimer l'étiquette** et **Requête HTTP** ne sont pas exécutées. Le traitement de l'action s'arrêtera à **Définir l'imprimante**, Automation Manager affichera que le déclencheur est en état d'erreur et le retour d'informations du déclencheur (s'il est activé) donnera : « mauvaise sélection d'imprimante/imprimante inaccessible ».



Mais dans ce cas particulier, il ne faut pas utiliser le retour synchronisé, envoyé automatiquement quand il est activé dans le déclencheur supportant le retour synchrone. Le retour d'informations doit être fait de façon asynchrone avec l'action **Requête HTTP** après la création du travail d'impression (ou non). À la fin du processus d'impression, mettre une application à jour avec son état. Pour ce faire, envoyer l'application en tant que message formaté HTTP.

Dans ce cas, l'action **Requête HTTP** doit être exécutée sans tenir compte du succès de toutes les actions situées au-dessus d'elle dans la liste. Il faut alors activer l'option **Ignorer les échecs** pour toutes les actions qui sont au-dessus de l'action **Requête HTTP**. L'option est disponible dans les options **Exécution et traitement d'erreurs** d'une action.



Si une action particulière échoue, NiceLabel Automation commencera l'exécution de l'action suivante au niveau suivant de la hiérarchie.

Exemple

Si l'action **Définir l'imprimante** au niveau 1.1 échoue, l'exécution ne continue pas avec l'action **Imprimer l'étiquette** au niveau 1.2 car elle échouera probablement aussi. Elle continue avec l'action **Requête HTTP** au niveau 2 car c'est l'action qui suit au niveau suivant de la hiérarchie.

La même logique peut être implémentée pour les actions répétitives, telles qu'**Utiliser un Filtre de Données**, **Boucler** et **Pour chaque enregistrement**. Elles sont répétées pour toutes les actions de la liste. Si le traitement d'une action échoue pour n'importe quelle raison, par défaut NiceLabel Automation arrêtera le traitement de toutes les autres actions et rapportera une erreur. Avec l'option **Ignorer l'échec** activée, le traitement s'arrêtera pour l'action qui a échoué mais NiceLabel Automation continuera pour l'action suivante. Dans tous les cas, l'erreur sera rapportée.

7.4. Mode d'impression synchrone

INFO NIVEAU DE PRODUIT

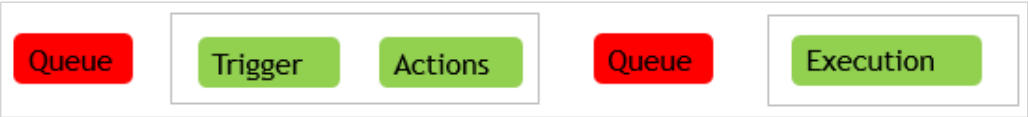
Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

7.4.1. Mode d'impression asynchrone

Par défaut, le mode d'opération de NiceLabel Automation est asynchrone. Le mode asynchrone est une forme d'impression lors de laquelle un déclencheur envoie des données à imprimer et ferme la connexion avec le sous-système d'impression. Le déclencheur n'attend pas les résultats du processus d'impression et ne reçoit pas de retour d'informations. Dès que les données ont été envoyées, le déclencheur est prêt à recevoir un nouveau flux de données.

Le mode asynchrone amplifie les performances du déclencheur et augmente le nombre de déclencheurs qui peuvent être traités dans le temps. Chaque processus d'impression est en mémoire tampon, c'est là où le déclencheur fournit les requêtes d'impression. La mémoire tampon s'adapte aux pics des déclencheurs et s'assure qu'aucune donnée n'est perdue.

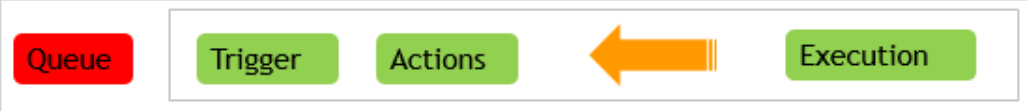
Si une erreur survient durant le traitement, elle est enregistrée dans Automation Manager (et NiceLabel Control Center, si utilisé), mais le déclencheur n'en est pas informé. Avec Automation en mode d'impression asynchrone, il est impossible de définir l'exécution d'actions conditionnelles, si l'exécution du déclencheur est en erreur.



7.4.2. Mode d'impression synchrone

Contrairement au mode asynchrone, le mode synchrone n'arrête pas la connexion lorsque le processus d'impression est initié. Dans ce mode, le déclencheur envoie les données d'impression et reste connecté au sous-système d'impression tant qu'il exécute les actions. Quand le processus d'impression se termine (avec succès ou avec une erreur), le déclencheur reçoit le retour d'informations concernant l'état.

Cette information peut être utilisée dans les actions définies dans le même déclencheur pour décider d'exécuter une autre action si une erreur survient. L'état du travail d'impression peut aussi être renvoyé à l'application qui a fourni les données. Pour plus d'informations, consulter l'article Retour d'informations sur le travail d'impression.



Exemple

L'information sur l'état de l'impression peut être renvoyée à l'application ERP qui a fourni les données.

Activer le mode d'impression synchrone

Le mode synchrone est définissable par déclencheur. Pour activer le mode synchrone dans un déclencheur, effectuer les opérations suivantes :

1. Ouvrir les propriétés du déclencheur.
2. Aller à l'onglet **Paramètres**.
3. Sélectionner **Autre**.
4. Dans la section **Commentaires du moteur d'impression,** activer l'option **Impression supervisée**.

NOTE

Quand le déclencheur fonctionne en mode d'impression synchrone, il communique avec un seul processus d'impression. L'activation du mode d'impression synchrone garantit la méthode FIFO de manipulation des événements en sortie (impression). Par défaut, le traitement en mode multi-cœur ne peut pas garantir l'ordre d'impression.

Par défaut, le traitement en mode multi-cœur ne permet pas de garantir l'ordre d'impression. En changeant les paramètres d'impression multi threads par défaut et en limitant les requêtes simultanées, vous pouvez assurer que le mode FIFO :

1. Ouvrir le fichier **`product.config`** dans un éditeur de texte.
   Le fichier est ici :

   ```
   %PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
   ```

2. Définir les valeurs des éléments **MaxConcurrentPrintProcesses** et **MaxConcurrentSessionPrintProcesses** sur **1**.

   ```
   <configuration>
       <IntegrationService>
           <MaxConcurrentPrintProcesses>1</
   MaxConcurrentPrintProcesses>
           <MaxConcurrentSessionPrintProcesses>1</
   MaxConcurrentSessionPrintProcesses>
       </IntegrationService>
   </configuration>
   ```

3. Enregistrer le fichier. met automatiquement à jour le service avec le nouveau nombre de threads d'impression.
4. Ouvrir Builder et aller dans vos propriétés de déclencheur.
5. Aller dans l'onglet **Exécution**.

6. Dans la section **Requêtes simultanées**, définir **Nombre d'exécutions d'actions simultanées**sur **1**.

7.5. Retour d'informations sur le travail d'impression

L'application qui fournit les données d'impression à NiceLabel Automation veut parfois recevoir des informations concernant l'état des travaux d'impression. Le retour d'informations peut être simple tel que « Tout OK » dans le cas où le travail d'impression est réussi, ou plus détaillé en cas de problème. Pour des raisons de performances, par défaut NiceLabel Automation NiceLabel Automation désactive les possibilités de retour d'informations. Ainsi, l'exécution de l'impression est plus rapide car le déclencheur ne s'occupe pas du processus d'exécution d'impression. Les erreurs sont enregistrées dans le journal de la base de données, mais le déclencheur ne va pas les traiter.

Cette méthode permet aussi d'envoyer des informations concernant les autres données que le déclencheur peut collecter. Par exemple, l'état des imprimantes réseau, le nombre de travaux d'impression dans le spouleur, la liste d'étiquettes dans un dossier, la liste de variables dans le fichier d'étiquette spécifié, etc.

NOTE

Pour activer le retour d'informations par le moteur d'impression, il faut activer le mode d'impression synchrone. Pour plus d'informations, consulter l'article Mode d'impression synchrone.

Activer le retour d'informations sur le travail d'impression en utilisant l'une des deux méthodes disponibles.

Le déclencheur renvoie les informations concernant l'état des travaux d'impression (Retour synchrone)
Certains déclencheurs ont une capacité de retour d'informations intégrée. Quand le mode d'impression synchrone est activé, le déclencheur connaît en interne l'état du travail d'impression. Le client peut envoyer les données au déclencheur, garder la connexion ouverte et attendre le retour. Pour utiliser cette méthode de retour d'informations, sélectionner utiliser un type de déclencheur qui supporte le retour d'informations.

Si une erreur survient dans une des actions, la variable interne **`LastActionErrorDesc`** contient le message d'erreur détaillé. La valeur de ce message peut être utilisée telle quelle ou personnalisée.

Pour plus d'informations, consulter les détails des déclencheurs respectifs.

- Déclencheur Web Service : Par défaut, ce déclencheur supporte le retour d'informations. Le document WSDL (Web Service Description Language) décrit les détails concernant l'interface Service Web et indique comment activer le retour d'informations. Utiliser la réponse par défaut qui envoie la description d'erreur en cas de défaillance de l'action d'impression. Ou personnaliser la réponse et renvoyer le contenu d'une variable. La variable peut contenir des données diverses, y compris un aperçu de l'étiquette ou le travail d'impression (données binaires).
- Déclencheur serveur HTTP : Par défaut, ce déclencheur supporte le retour d'informations. NiceLabel Automation va utiliser le code de réponse HTTP standard pour indiquer l'état du travail d'impression. La réponse HTTP peut être personnalisée pour renvoyer le contenu d'une variable. La variable peut contenir des données diverses, y compris un aperçu de l'étiquette ou le travail d'impression (données binaires).
- Déclencheur serveur TCP/IP : Ce déclencheur supporte le retour d'informations, mais pas automatiquement. Pour qu'il renvoie des informations, il faut configurer le client qui fournit les données pour ne pas interrompre la connexion après l'envoi des données. Après la fin du processus d'impression, l'action suivante dans la liste peut être Envoyer les données au port TCP/IP avec le paramètre Répondre à l'expéditeur activé. Les informations peuvent être renvoyées sur la connexion restée ouverte.

L'action fournit le renvoi d'informations d'état du travail d'impression (retour asynchrone)

Pour les déclencheurs qui ne supportent pas le retour d'informations en interne ou pour envoyer des messages d'information durant le traitement du déclencheur, il est possible de définir une action qui renverra les informations à une destination donnée. Dans ce cas, l'application fournissant les données peut fermer la connexion dès que les données pour le déclencheur ont été fournies.

Exemple

Le déclencheur TCP/IP est utilisé pour collecter les données. Le client a terminé la connexion immédiatement après l'envoi des données, donc nous ne pouvons pas répondre sur la même connexion. Dans ce cas, les informations sont renvoyées sur un autre canal. Vous pouvez configurer une des actions de connexion sortante, telle que Exécuter l'instruction SQL, Ouvrir Document/Programme, Requête HTTP, Envoyer les données au port TCP/IP et autres. Il faut alors placer cette action dans l'action Imprimer l'étiquette.

Pour renvoyer des informations sur un statut spécifique, tel que « erreur survenue », utiliser l'une des méthodes suivantes.

- **Utiliser la condition de l'action :** L'état du travail d'impression est exposé dans deux variables internes (`LastActionErrorID` et `LastActionErrorDesc`). La première contient l'ID de l'erreur ou la valeur `0` en l'absence d'erreur. La seconde contient un message d'erreur détaillé. Les valeurs de ces variables sont utilisables dans les conditions des actions à exécuter en cas d'erreur. Par exemple, vous pouvez utiliser l'action **Requête HTTP** à la fin de l'impression. L'action renverrait un retour d'informations en cas d'erreur. Pour activer ce retour, effectuer les opérations suivantes :

  1. Ouvrir les propriétés du déclencheur.
  2. Dans le groupe de ruban **Variable**, cliquer sur **Variables internes** et activer la variable `ActionLastErrorID`.
  3. Aller sur l'onglet Actions.
  4. Ajouter l'action **Envoyer les données à HTTP**.
  5. Dans les propriétés de l'action, étendre **Afficher les options d'exécution et de traitement d'erreurs**.
  6. Pour **Condition**, saisir ce qui suit. L'action ayant cette condition s'exécute seulement si l'erreur survient et si `LastErrorActionID` contient l'ID de l'erreur (toute valeur plus grande que `0`). Par défaut, les conditions s'exécutent en utilisant la syntaxe VB Script.

     ```
     ActionLastErrorID > 0
     ```

  7. Vous devez aussi activer l'option **Ignorer l'échec** sur chaque action qui va échouer. Cela prévient Automation de ne pas arrêter complètement l'exécution des actions, mais de continuer par l'action suivante au même niveau hiérarchique.

> **NOTE**
> Pour plus d'informations, consulter l'article Traitement d'erreur.

- **Utiliser l'action Essayer :** L'action Essayer élimine le besoin de coder les conditions. L'action fournit deux espaces réservés. L'espace réservé **Do** (faire) contient les actions à exécuter. Si une erreur survient, l'exécution s'interrompt et les actions dans l'espace réservé **Pour erreur** sont exécutées. Les actions de connectivité du trafic en sortie de cet espace réservé sont utilisées pour renvoyer les informations sur le statut du travail d'impression. Pour plus d'informations, consulter l'article Essayer.

7.6. Exclure les imprimantes de l'impression automatisée

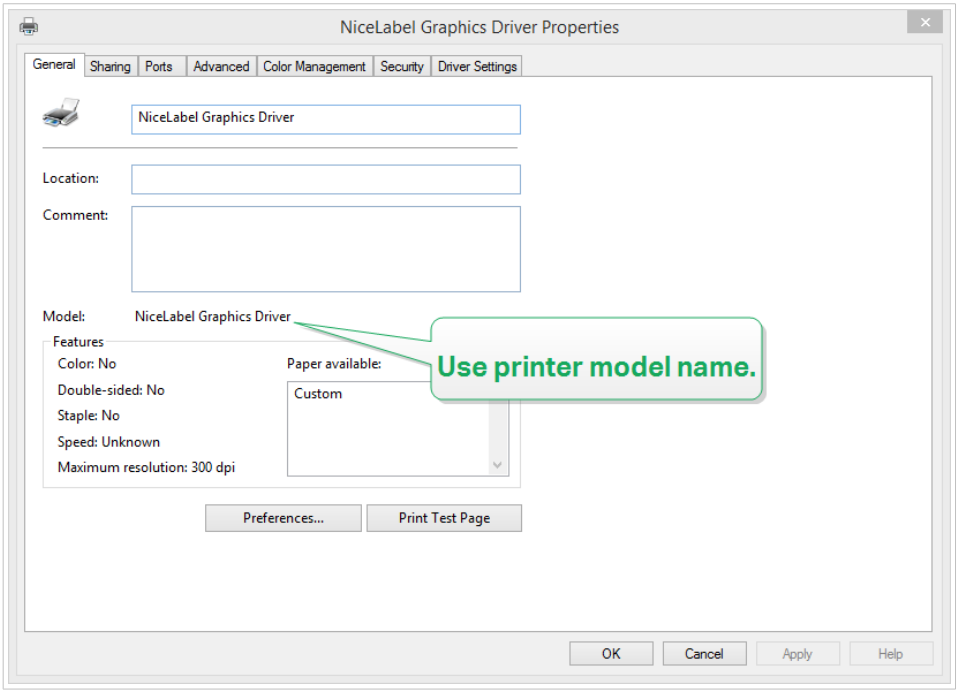
Dans certains cas, votre environnement d'impression vous oblige à exclure certaines imprimantes du processus d'impression automatisée. Les raisons possibles pour lesquelles vous devez exclure les imprimantes de l'impression automatisée reposent sur la politique d'impression de votre entreprise ou les limites de votre licence.

Par défaut, Automation empêche l'impression automatisée d'utiliser des imprimantes de fichiers telles que Microsoft Shared Fax Driver, Microsoft Print To PDF, Microsoft XPS Document Writer et autres. Ces imprimantes de fichiers obligent l'utilisateur à sélectionner manuellement l'emplacement de leurs impressions. L'intervention manuelle de l'utilisateur provoque l'arrêt du moteur d'impression, rapportant une erreur.

Pour empêcher Automation d'utiliser des imprimantes spécifiques dans les configurations d'exécution :

> **NOTE**
> En spécifiant les imprimantes à exclure de l'impression automatisée dans le fichier **`product.config`**, vous devez aussi lister explicitement vos imprimantes de fichiers.

1. Ouvrir le fichier **`product.config`** dans un éditeur de texte.
   Le fichier est ici :
   `%PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config`
2. Créer une copie de sauvegarde du fichier **`product.config`**.
3. Automation utilise deux paramètres pour vérifier quel modèle et quel port de l'imprimante doivent être exclus de l'impression automatisée. Ajouter ces paramètres à votre fichier **`product.config`**.
   - **`/IntegrationService/DisabledPrinterDrivers`** et saisir les modèles d'imprimantes que vous voulez exclure de l'impression automatisée.
   - **`/IntegrationService/DisabledPrinterDrivers`** et saisir les ports que vous voulez exclure de l'impression automatisée.




```
<configuration>
    <IntegrationService>
        <DisabledPrinterDrivers>NiceLabel Graphics
Driver,NicePrinter1200dpi</DisabledPrinterDrivers>
        <DisabledPrinterPorts>LPT1,LPT2</DisabledPrinterPorts>
    </IntegrationService>
</configuration>
```

4. Après avoir défini les modèles et ports d'imprimantes dans le fichier **`product.config`**, vous pouvez toujours exécuter vos configurations Automation, mais vos paramètres mis à jour empêchent l'impression sur les imprimantes listées. Automation rapporte une erreur si ces imprimantes font partie des configurations d'exécution.

> **NOTE**
> L'impression automatisée ne s'arrête pas si les imprimantes exclues sont utilisées dans l'action Rediriger l'impression vers un fichier.

7.7. Utiliser le mode d'impression Stocker/Rappeler

Le mode d'impression Stocker/Rappeler optimise le processus d'impression. Il augmente le temps de réponse de l'imprimante en diminuant le nombre de données qui doivent lui être envoyées durant les travaux d'impression répétitifs.

Avec le mode d'impression Stocker/Rappeler activé, NiceLabel Automation n'a pas besoin de renvoyer les données d'étiquette complètes pour chaque impression. Les masques d'étiquettes sont enregistrés dans

la mémoire de l'imprimante. Les objets fixes sont enregistrés comme tels, alors que des espaces réservés sont définis pour les objets variables. NiceLabel Automation n'envoie que les données pour les objets variables de l'étiquette et les commandes de rappel. L'imprimante affecte les données reçues aux espaces réservés de l'étiquette enregistrée et imprime l'étiquette (en la rappelant de la mémoire). Typiquement, quelques octets de données sont envoyés à l'imprimante, comparé à plusieurs kilo octets dans le cas d'une impression normale.

L'action comporte deux processus :

- **Stocker l'étiquette :** Durant ce processus, l'application crée une description du masque d'étiquette formaté dans le langage spécifique de commande de l'imprimante. Quand elle a terminé, l'application envoie le fichier de commande créé à la mémoire de l'imprimante qui l'enregistre. L'éditeur d'étiquettes ou NiceLabel Automation peuvent stocker l'étiquette en utilisant l'action Stocker l'étiquette dans l'imprimante.

  NOTE
  Le mode d'impression Stocker/Rappeler doit être défini dans les propriétés de l'étiquette pour pouvoir être enregistré dans l'imprimante.

- **Rappeler (imprimer) l'étiquette :** Une étiquette stockée dans la mémoire de l'imprimante s'imprime immédiatement. Lors de l'utilisation du processus de Rappel, NiceLabel Automation crée un autre fichier de commande pour dire à l'imprimante quelle étiquette elle doit imprimer de sa mémoire. La quantité réelle de données envoyées à l'imprimante dépend de la situation en cours. Pour des étiquettes fixes sans contenus variables, le fichier de commande Rappel ne contient que la commande de rappel de l'étiquette. Pour les étiquettes contenant des champs variables, le fichier de commande inclut les valeurs de ces variables et la commande de rappel.
  Pour rappeler une étiquette depuis NiceLabel Automation, utiliser simplement les actions d'impression. Quand elle est exécutée, l'action analyse l'étiquette et active le mode d'impression correspondant : mode d'impression normal ou avec Rappel, comme défini dans l'étiquette.

AVERTISSEMENT
Avant d'activer ce mode, vérifier que le pilote d'imprimante sélectionné est bien celui de l'imprimante d'étiquettes. Toutes les imprimantes n'ont pas la possibilité d'utiliser le mode d'impression Stocker/Rappeler. Le pilote de l'imprimante pour laquelle l'étiquette a été créée dans l'éditeur d'étiquettes doit également être installé sur l'appareil sur lequel NiceLabel Automation tourne.

7.8. Cluster haute disponibilité (Fail-over)

INFO NIVEAU DE PRODUIT
La fonctionnalité de cet élément est disponible dans LMS Enterprise.

NiceLabel Automation peut utiliser le cluster haute disponibilité de Microsoft (failover). Un cluster est un groupe d'ordinateurs indépendants qui travaillent ensemble pour augmenter la disponibilité d'impression d'étiquettes avec NiceLabel Automation. Les serveurs en cluster (appelés nœuds) sont connectés ensemble par des câbles et des logiciels. Si un ou plusieurs des nœuds du cluster est défaillant, les autres nœuds reprennent le service (ce processus est connu sous le nom de basculement). De plus, les rôles repris sont surveillés de façon pro-active pour vérifier qu'ils fonctionnent convenablement. S'ils ne fonctionnent pas, ils sont redémarrés ou déplacés vers un autre nœud. Les clients qui envoient des données se connectent à l'adresse IP appartenant au cluster entier, pas à l'adresse IP individuelle du nœud.

Pour activer NiceLabel Automation dans le but d'utiliser un cluster haute disponibilité, effectuer les opérations suivantes :

- Activer la fonctionnalité Microsoft Failover Clustering sur vos Serveurs Windows.
- Installer NiceLabel Automation sur chaque nœud.
- Activer le support de reprise en nœud (cluster failover) dans les caractéristiques NiceLabel Automation de chaque nœud.
  Effectuer les opérations suivantes :
  1. Ouvrir **Fichier** > **Options** > **Automation**.
  2. Sous le groupe **Utilisation en cluster**, activer **Support de reprise en nœud**.
  3. Rechercher le dossier, situé en dehors des deux nœuds, mais accessible avec les droits d'accès complets au logiciel NiceLabel Automation . Les fichiers système importants, dont les deux nœuds ont besoin, doivent être copiés dans ce dossier.
- Configurer le cluster pour démarrer NiceLabel Automation sur le second nœud quand le nœud maître est en panne.

Consultez la description détaillée de la configuration du cluster de basculement Automation dans notre Article de la base de connaissances.

7.9. Cluster de répartition des charges

INFO NIVEAU DE PRODUIT

La fonctionnalité de cet élément est disponible dans LMS Enterprise.

NiceLabel Automation est compatible avec le cluster de répartition des charges de Microsoft. Un cluster de répartition de charges est un groupe d'ordinateurs indépendants qui travaillent ensemble pour augmenter la haute disponibilité d'impression d'étiquettes au travers de NiceLabel Automation. Les serveurs en cluster (appelés nœuds) sont connectés ensemble par des câbles et des logiciels. Les requêtes entrantes d'impression d'étiquettes sont distribuées entre tous les nœuds du cluster. Les clients qui envoient des données se connectent à l'adresse IP appartenant au cluster, pas à l'adresse IP individuelle du nœud.

NOTE

Vous pouvez utiliser les déclencheurs basés sur TCP/IP avec le cluster de répartition des charges. Ces déclencheurs sont #UUID-f4d9621f-a506-aa16-4740-7b9fb962f0d3Déclencheur serveur TCP/IP, Déclencheur serveur HTTP, Déclencheur Web Service, et Déclencheur de Cloud.

Pour activer la répartition des charges de NiceLabel Automation, effectuer les opérations suivantes :

- Activer la fonctionnalité Microsoft Load-balancing Clustering dans vos Serveurs Windows.
- Installer NiceLabel Automation sur chaque nœud.
- Chargez les mêmes fichiers de configuration dans Automation Manager sur chaque nœud.

8. Comprendre les structures de données

Cette section démontre les structures de données basiques qui sont fréquemment utilisées dans les scénarios d'Automation. Si vous travaillez avec plusieurs fichiers de données, nous devons analyser leur structure, extraire les données pertinentes des champs d'intérêt et les imprimer sur les étiquettes. Chaque cas listé ci-dessous est utilisé dans les exemples de configuration inclus dans le pack d'installation de Automation.

- Base de données texte
- CSV composé
- Fichiers binaires
- Données existantes
- Fichiers de commande
- Données XML
- Données JSON

8.1. Fichiers binaires

Les fichiers binaires sont des fichiers qui ne contiennent pas seulement du texte mais aussi des caractères binaires tels que les codes de contrôle (caractères sous le code ASCII 32). Filtre de données non structurées supporte les caractères binaires. Les caractères binaires permettent de définir des positions de champs, et sont aussi utilisables comme valeurs de champs.

Exemple typique : l'exportation de données d'un système existant, dans lequel les données d'étiquettes sont délimitées par un caractère **<FF>**, saut de page.

8.1.1. Exemple

Dans ce cas, le déclencheur Automation capture le flux d'impression. La section surlignée en jaune doit être extraite du flux et envoyée à une imprimante différente. Le filtre est configuré pour rechercher le **<FF>** en position de fin de champ.

```
<ESC>%-12345X@PJL USTATUSOFF
@PJL INFO STATUS
@PJL USTATUS DEVICE=ON
<ESC>%-12345X<ESC>%-12345X




^^02^L
^^02^O0270
D11
H15
PE
SE
Q0001
131100000300070001-001-001
1e420550075005000010010 19
1322000001502859
W
E
<FF><ESC>%-12345X<ESC>%-12345X@PJL USTATUSOFF
<ESC>%-12345X
```

8.2. Fichiers de commande

Les fichiers de commande sont des fichiers texte contenant des commandes qui seront exécutées une par une, de haut en bas. NiceLabel Automation supporte les fichiers de commande originaux, ainsi que les fichiers de commande XML Oracle et SAP. Pour plus d'informations, consulter l'article Caractéristiques des fichiers de commande,Caractéristiques Oracle XML,Caractéristiques SAP AII XML.

8.2.1. Exemple

L'étiquette **`label2.nlbl`** va s'imprimer sur l'imprimante **`CAB A3 203DPI`**.

```
LABEL "label2.nlbl"
SET code="12345"
SET article="FUSILLI"
SET ean="383860026501"
SET weight="1,0 kg"
PRINTER "CAB A3 203DPI"
PRINT 1
```

8.3. CSV composé

Le CSV composé est un fichier texte qui contient des données dans deux structures, dans une structure CSV standard et dans une entête multilignes qui utilise une structure non standard. Le contenu d'un fichier CSV composé ne peut pas être analysé en utilisant un seul filtre. Pour analyser les données dans les deux structures, configurer deux filtres distincts :

- Filtre de texte structuré pour les champs dans une structure CSV
- Filtre de données non structurées pour les champs dans une structure d'entête non standard.

Les configurations incluant un fichier CSV composé requiert deux actions qui exécutent les deux filtres sur les données reçues.

8.3.1. Exemple

Les données de la ligne 3 jusqu'à la fin du document ont une structure CSV et sont analysées par le filtre de texte structuré. Les données des 2 premières lignes n'ont pas de structure particulière et sont analysées par le filtre de texte non structuré.

```
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"
```

8.4. Données existantes

Les données existantes sont des exportations non structurées ou semi-structurées des applications existantes. Ces exportations n'utilisent pas de structures de données CSV ou XML. Pour extraire les données pertinentes de ces fichiers, utiliser Filtre de données non structurées et définir les positions des champs d'intérêt. Le filtre dans Automation extrait les valeurs des champs et les rend disponibles pour l'impression sur des étiquettes.

8.4.1. Exemple

Le fichier ci-dessous ne suit pas de règle de structure. Chaque champ doit être configuré manuellement.

```
HAWLEY     ANNIE     ER12345678 ABC     XYZ
                9876543210
```

```
PRE OP     07/11/12     F 27/06/47     St. Ken Hospital     3

G015 134 557 564 9     A- 08/11/12     LDBS F-         PB     1
G015 134 654 234 0     A- 08/11/12     LDBS F-         PB     2
G015 134 324 563 C     A- 08/11/12     LDBS F-         PB     3



                Antibody Screen: Negative
    Store Sample :
    SAMPLE VALID FOR 24 HOURS, NO TRANSFUSION HISTORY SUPPLIED

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

8.5. Base de données texte

La base de données texte est un alias donné aux fichiers texte à champs structurés, tels que CSV (texte séparé par une virgule), ou fichier à champs de largeur fixe. Dans tous les cas, cliquer sur le bouton **Importer la structure de données** et suivre les instructions de l'interface pour importer les champs. Quand dans un fichier de données avec une structure délimitée, le nombre de champs varie d'une copie à l'autre, activer l'élément **Structure dynamique** et laisser NiceLabel Automation traiter l'extraction de données et le mappage automatique avec les variables. Pour plus d'informations, consulter l'article Activer la structure dynamique.

8.5.1. Exemple

- **Fichier avec champs délimités :** La première ligne du fichier contient les noms de champs que le filtre peut importer.

```
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
```

- **Fichier avec champs de largeur fixe :** Les champs contiennent un nombre fixe de caractères.

```
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
```

8.6. Données XML

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

XML signifie eXtensible Markup Language. Les balises XML ne sont pas prédéfinies, vous êtes libre de définir vos propres balises pour décrire les données. XML est conçu pour être auto-descriptif.

La structure XML est définie par des éléments, des attributs (et leurs valeurs), et texte (élément texte).

8.6.1. Exemples

Oracle XML

Le traitement d'Oracle XML fait partie intégrante du logiciel. Pas besoin de configurer de filtres pour extraire les données, il suffit de lancer l'action intégrée Exécuter le Fichier de Commande Oracle XML. Pour plus d'informations concernant le journal, consulter l'article Caractéristiques Oracle XML.

```xml
<?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>
```

XML en général

Si le logiciel n'est pas compatible avec la structure XML, définir le filtre XML et définir les règles d'extraction de données. Pour plus d'informations, voir l'article Comprendre les filtres.

```xml
<?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

Le traitement de NiceLabel XML fait partie intégrante du logiciel. Pas besoin de configurer de filtres pour extraire les données, il suffit de lancer l'action intégrée Exécuter un Fichier de Commande. Pour plus d'informations concernant le journal, consulter l'article Fichier de commande XML.

```
<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>
```

8.7. Données JSON

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise** et **LMS Pro** .

JavaScript Object Notation (JSON) est un format de fichier ouvert standard. JSON utilise le texte lisible pour transmettre des objets de données consistant en des paires nom-valeur, et les types de données d'une plage (ou toute valeur en série). JSON est un format de données très commun utilisé pour la communication asynchrone navigateur-serveur, incluant un remplacement pour XML.

Il existe plusieurs ressources en ligne qui décrivent les similitudes et les différences entre JSON et XML. Le tableau ci-dessous en décrit une partie :

JSON	XML
Signifie JavaScript Object Notation	Signifie Extensible Markup Language

JSON	XML
Basé sur le langage JavaScript.	Dérivé de SGML.
Une manière de représenter les objets.	Un langage de balises utilisées pour représenter des objets de données.
Ne prend pas en charge les noms d'espace.	Prend en charge les noms d'espace.
Prend en charge le spectre.	Ne prend pas en charge le spectre.
Ses fichiers sont très faciles à lire comparé à XML.	Ses documents sont très difficiles à lire et à interpréter.
N'utilise pas de balise de fin.	Utilise des balises de début et de fin.
Moins sécurisé.	Plus sécurisé que JSON.
Ne prend pas en charge les commentaires.	Prend en charge les commentaires.
Ne prend en charge que l'encodage UTF-8.	Prend en charge divers encodages.

Source : https://www.geeksforgeeks.org/difference-between-json-and-xml/

Exemples

```
{
  "DELIVERYNOTE": {
    "LIST_CUSTOMER_INFO": {
      "CUSTOMER_INFO": {
        "CUSTOMER_NAME": "Customer A",
        "CUSTOMER_STREET_ADDRESS": "Test St",
        "CUSTOMER_POST_ADDRESS": "1234, Test City",
        "CUSTOMER_NUMBER": "1234",
        "CURRENCY": "EUR",
        "DELIVERY_METHOD": "Express delivery",
        "EDI_INFORMATION": "EDI",
        "ORDER_TYPE": "CSO",
        "ORDER_NUMBER": "123",
        "LIST_ITEM": {
          "ITEM": [
            {
              "ARTICLE_NUMBER": "0001",
              "ARTICLE_NAME": "Collins Complete Woodworker's Manual",
              "PRICE": "23.3"
            },
            {
              "ARTICLE_NUMBER": "0002",
              "ARTICLE_NAME": "Be Careful What You Wish For (Clifton
Chronicles)",
              "PRICE": "16.6"
            },
            {
              "ARTICLE_NUMBER": "0003",
              "ARTICLE_NAME": "Mockingjay (part III of Hunger Games Trilogy)",
              "PRICE": "25.0"
            }
          }
        }
      }
    }
  }
}
```

```
{
    "NICELABEL_JOB": {
      "TIMESTAMP": "20130221100527.788134",
      "USER": "PGRI",
       IT_LABEL_DATA": {
        "LBL_NAME": "goods_receipt.nlbl",
        "LBL_PRINTER": "Production01",
```

```
        "LBL_QUANTITY": "1",
        "MAKTX": "MASS ONE",
        "MATNR": "28345",
        "MEINS": "KG",
        "WDATU": "19.01.2012",
        "QUANTITY": "1",
        "EXIDV": "012345678901234560"
      }
    }
  }
}
```

9. Référence et résolution de problèmes

9.1. Types de fichiers de commande

9.1.1. Caractéristiques des fichier de commande

Les fichiers de commande contiennent des instructions pour chaque processus d'impression. Ces instructions sont exprimées avec les commandes NiceLabel. Les commandes sont exécutées une par une du début à la fin du fichier. Les fichiers supportent le formatage Unicode, ils peuvent donc comporter des données multilingues. Les fichiers de commande se présentent sous trois formes.

9.1.2. Fichier de commande CSV

Les commandes disponibles dans les fichiers de commande CSV forment un sous-ensemble des commandes NiceLabel. Les commandes suivantes sont disponibles : **LABEL**, **SET**, **PORT**, **PRINTER** et **PRINT**.

CSV signifie valeurs séparées par une virgule. C'est un fichier texte dans lequel les valeurs sont séparées par une virgule (`,`). Le fichier texte peut contenir des valeurs Unicode (important pour les données multilingues). Chaque ligne d'un fichier de commande CSV contient les commandes pour une action d'impression d'étiquette.

La première ligne du fichier de commande CSV doit contenir les noms des commandes et des variables. L'ordre des noms de commandes n'est pas important, mais toutes les données d'un même flux d'impression doivent avoir la même structure. Les paires de variables **`nom-valeur`** sont extraites automatiquement et envoyées à l'étiquette. Si la variable d'un nom du CSV n'existe pas dans l'étiquette, aucun message d'erreur n'est affiché.

9.1.2.1. Exemple de fichier de Commande CSV

L'exemple présente une vue structurelle des champs utilisables dans le fichier de commande CSV.

```
@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
```

Caractéristiques du fichier de Commande CSV

Les commandes de la première ligne de données doivent être exprimées avec le caractère (@). Les champs sans @ au début sont des noms de variables. Ces champs sont extraits avec leurs valeurs en tant que paires **`nom-valeur`**.

- **@Label:** Spécifie le nom d'étiquette à utiliser. Il est conseillé d'inclure le chemin et le nom du fichier. Il faut que l'utilisateur du service puisse accéder au fichier. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation. C'est un champ obligatoire.
- **@Printer:** Spécifie l'imprimante à utiliser. Il remplace l'imprimante définie dans l'étiquette. Il faut que l'utilisateur du service puisse accéder à l'imprimante. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées. C'est un champ facultatif.
- **@Quantity:** Spécifie le nombre d'étiquettes à imprimer. Valeurs possibles : valeur numérique, VARIABLE ou UNLIMITED. Pour plus d'informations, consulter l'article Imprimer une étiquette. C'est un champ obligatoire.
- **@Skip:** Spécifie le nombre d'étiquettes à sauter au début de la première page imprimée. Cette fonctionnalité est utile quand la feuille d'étiquettes utilisée est déjà partiellement imprimée. C'est un champ facultatif.
- **@IdenticalCopies:** Spécifie le nombre de copies d'étiquettes à imprimer pour chaque étiquette unique. Cette fonctionnalité est utile à l'impression d'étiquettes contenant des données d'une base de données, pour utiliser des compteurs et pour faire des copies d'étiquettes. C'est un champ facultatif.
- **@NumberOfSets:** Spécifie combien de fois le processus d'impression complet doit être répété. Chaque jeu d'étiquettes est égal à une seule occurrence du processus d'impression. C'est un champ facultatif.
- **@Port:** Spécifie le nom du port pour l'imprimante. Il peut remplacer le port par défaut spécifié dans le pilote d'imprimante. Il permet aussi de rediriger l'impression vers un fichier. C'est un champ facultatif.
- **Autres noms de champs :** Tous les autres champs définissent les noms des variables de l'étiquette. Le contenu des champs est enregistré dans la variable ayant le même nom avec sa valeur.

9.1.3. Fichier de commande JOB

Le fichier de commande JOB est un fichier texte contenant les commandes NiceLabel. Les commandes s'exécutent du haut vers le bas. Les commandes commencent habituellement par LABEL (pour ouvrir l'étiquette), ensuite SET (pour régler la valeur de variable), et finalement PRINT (pour imprimer l'étiquette). Pour plus d'informations concernant les commandes disponibles, consulter l'article Utilisation de commandes personnalisées.

9.1.3.1. Exemple de fichier de Commande JOB

Ce fichier JOB ouvre **`label2.nlbl`**, définit les valeurs des variables et imprime une seule étiquette. Comme aucune commande PRINTER n'est utilisée pour rediriger l'impression, l'étiquette est imprimée en utilisant l'imprimante définie dans l'étiquette.

```
LABEL "label2.nlbl"
SET code="12345"
SET article="FUSILLI"
SET ean="383860026501"
SET weight="1,0 kg"
PRINT 1
```

9.1.4. Fichier de commande XML

Les commandes disponibles dans les fichiers de commande CSV sont un sous-ensemble des commandes NiceLabel. Les commandes suivantes sont disponibles : **LOGIN**, **LABEL**, **SET**, **PORT**, **PRINTER**, **SESSIONEND**, **SESSIONSTART** et **SESSIONPRINT**. La syntaxe de ces commandes varie si elles sont utilisées dans un fichier XML.

L'élément de base du fichier de commande XML est **`<Nice_Commands>`**. L'élément qui doit suivre est **`<Label>`**. L'élément spécifie l'étiquette à utiliser.

Il y a deux méthodes pour lancer l'impression d'étiquettes : imprimer les étiquettes normalement en utilisant l'élément **`<Print_Job>`**, ou imprimer les étiquettes en sessions en utilisant l'élément **`<Session_Print_Job>`**. Il est aussi possible de changer l'imprimante sur laquelle les étiquettes s'impriment et de paramétrer les valeurs de variables.

9.1.4.1. Exemple de fichier de commande XML

L'exemple ci-dessous présente une vue structurelle des éléments et de leurs attributs tels qu'ils peuvent être utilisés dans un fichier XML.

```
<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>
```

Exemple de fichier de commande XML

Cette section contient la description de structure du fichier de commande XML. Il y a différents éléments qui contiennent des attributs. Certains attributs sont obligatoires, d'autres sont en option. Certains attributs peuvent seulement comporter des valeurs prédéfinies, d'autres peuvent avoir des valeurs personnalisées.

- **<Nice_Commands> :** Élément racine.
- **<Label>:** Spécifie le fichier d'étiquette à ouvrir. Si l'étiquette est déjà ouverte, elle ne se rouvrira pas. Le fichier d'étiquette doit être accessible depuis cet ordinateur. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées. Cet élément peut se trouver plusieurs fois dans le fichier de commande.
    - **Nom :** Cet attribut contient le nom d'étiquette. Mettre éventuellement le nom du chemin de l'étiquette. Obligatoire.
- **<Print_Job>:** L'élément qui contient les données d'un travail d'impression. Cet élément peut se trouver plusieurs fois dans le fichier de commande.
    - **Imprimante :** À utiliser pour remplacer l'imprimante définie dans l'étiquette. L'imprimante doit être accessible depuis cet ordinateur. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées. Optionnel.
    - **Quantité :** À utiliser pour spécifier le nombre d'étiquettes à imprimer. Valeurs possibles : valeur numérique, VARIABLE ou UNLIMITED. Pour plus d'informations concernant les paramètres, consulter l'article Imprimer l'étiquette. Obligatoire.
    - **Sauter :** À utiliser pour spécifier le nombre d'étiquettes à sauter au début. Élément utile pour imprimer des planches d'étiquettes sur une imprimante laser, quand les premières étiquettes sont déjà imprimées. Pour plus d'informations, consulter l'article Imprimer une étiquette. Optionnel.
    - **Job_name :** À utiliser pour spécifier le nom du travail d'impression. Le nom spécifié est visible dans le spouleur d'impression. Pour plus d'informations, consulter l'article Paramétrer le nom du travail d'impression. Optionnel.
    - **Print_to_file :** À utiliser pour spécifier le nom du fichier dans lequel il faut enregistrer les commandes de l'imprimante. Pour plus d'informations, consulter l'article Rediriger l'impression vers un fichier. Optionnel.
    - **Identical_copies :** À utiliser pour spécifier le nombre de copies à imprimer pour chaque étiquette. Pour plus d'informations, consulter l'article Imprimer une étiquette.
- **<Session_Print_Job> :** Élément qui contient les données et les commandes pour une ou plusieurs sessions. L'élément peut contenir un ou plusieurs éléments **`<Session>`**. Il qualifie les règles d'impression de la session. Cet élément peut être utilisé plusieurs fois dans le fichier de commande. Pour rechercher les attributs disponibles, voir ceux de l'élément **`<Print_Job>`**. Ils sont tous

utilisables, sauf l'attribut de quantité. Voir la description de l'élément **<Session>** pour déterminer comment spécifier la quantité d'étiquettes dans la session d'impression.

- **<Session> :** L'élément qui contient les données pour une session. En imprimant en sessions, toutes les étiquettes sont codées et envoyées à l'imprimante dans un seul travail d'impression.
  - **Quantité :** À utiliser pour spécifier le nombre d'étiquettes à imprimer. Valeurs possibles : valeur numérique, chaîne de caractères VARIABLE ou chaîne UNLIMITED. Pour plus d'informations concernant les paramètres, consulter l'article Imprimer l'étiquette. Obligatoire.
- **<Variable> :** L'élément qui assigne les valeurs aux variables de l'étiquette. Cet élément peut se trouver plusieurs fois dans le fichier de commande.
  - **Nom :** L'attribut qui contient le nom de variable. Obligatoire.

Schéma de Définition XML (XSD) pour le fichier de commande XML

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd"
elementFormDefault="qualified" xmlns="http://tempuri.org/XMLSchema.xsd"
xmlns:mstns="http://tempuri.org/XMLSchema.xsd" xmlns:xs="http://www.w3.org/
2001/XMLSchema">
    <xs:element name="nice_commands">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="label" maxOccurs="unbounded" minOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="print_job" maxOccurs="unbounded"
minOccurs="0">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="database"
maxOccurs="unbounded" minOccurs="0">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension
base="xs:string">
                                                        <xs:attribute
name="name" type="xs:string" use="required" />
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="table"
maxOccurs="unbounded" minOccurs="0">
                                            <xs:complexType>
                                                <xs:simpleContent>
```

```
                                                        <xs:extension
base="xs:string">
                                                            <xs:attribute
name="name" type="xs:string" use="required" />
                                                        </xs:extension>
                                                    </xs:simpleContent>
                                                </xs:complexType>
                                            </xs:element>
                                            <xs:element name="variable"
maxOccurs="unbounded" minOccurs="0">
                                                <xs:complexType>
                                                    <xs:simpleContent>
                                                        <xs:extension
base="xs:string">
                                                            <xs:attribute
name="name" type="xs:string" use="required" />
                                                        </xs:extension>
                                                    </xs:simpleContent>
                                                </xs:complexType>
                                            </xs:element>
                                        </xs:sequence>
                                        <xs:attribute name="quantity"
type="xs:string" use="required" />
                                        <xs:attribute name="printer"
type="xs:string" use="optional" />
                                        <xs:attribute name="skip"
type="xs:integer" use="optional" />
                                        <xs:attribute name="identical_copies"
type="xs:integer" use="optional" />
                                        <xs:attribute name="number_of_sets"
type="xs:integer" use="optional" />
                                        <xs:attribute name="job_name"
type="xs:string" use="optional" />
                                        <xs:attribute name="print_to_file"
type="xs:string" use="optional" />
                                        <xs:attribute name="print_to_file_append"
type="xs:boolean" use="optional" />
                                        <xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
                                    </xs:complexType>
                                </xs:element>
                                <xs:element name="session_print_job"
maxOccurs="unbounded" minOccurs="0">
                                    <xs:complexType>
                                        <xs:sequence>
                                            <xs:element name="database"
```

```
maxOccurs="unbounded" minOccurs="0">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension
base="xs:string">
                                                        <xs:attribute
name="name" type="xs:string" use="required" />
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="table"
maxOccurs="unbounded" minOccurs="0">
                                            <xs:complexType>
                                                <xs:simpleContent>
                                                    <xs:extension
base="xs:string">
                                                        <xs:attribute
name="name" type="xs:string" use="required" />
                                                    </xs:extension>
                                                </xs:simpleContent>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="session"
minOccurs="1" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element
name="variable" minOccurs="0" maxOccurs="unbounded">
                                                        <xs:complexType>
                                                            <xs:simpleContent>
                                                                <xs:extension
base="xs:string">

<xs:attribute name="name" type="xs:string" use="required"
/>
                                                                </
xs:extension>
                                                            </xs:simpleContent>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                                <xs:attribute name="quantity"
type="xs:string" use="required" />
                                            </xs:complexType>
                                        </xs:element>
```

```
                                    </xs:sequence>
                                    <xs:attribute name="printer"
type="xs:string" use="optional" />
                                    <xs:attribute name="skip"
type="xs:integer" use="optional" />
                                    <xs:attribute name="job_name"
type="xs:string" use="optional" />
                                    <xs:attribute name="print_to_file"
type="xs:string" use="optional" />
                                    <xs:attribute name="print_to_file_append"
type="xs:boolean" use="optional" />
                                    <xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name="name" type="xs:string"
use="required" />
                        <xs:attribute name="close" type="xs:boolean"
use="optional" />
                        <xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="quit" type="xs:boolean" use="required" />
        </xs:complexType>
    </xs:element>
</xs:schema>
```

9.1.5. Caractéristiques Oracle XML

Oracle définit le format XML pour permettre la compréhension du contenu XML, son analyse, et ensuite l'imprimer comme une étiquette. La Définition du Type de Document XML (DTD) définit les balises XML utilisées dans le fichier XML. Oracle génère les fichiers XML selon cette DTD et le logiciel tiers traduit l'XML selon cette DTD.

Pour exécuter un tel fichier de commande, utiliser l'action Exécuter le Fichier de Commande XML . Exécuter le Fichier de Commande XML3 .

9.1.5.1. XML DTD

L'exemple qui suit est le XML DTD utilisé pour la formation de l'XML pour les formats synchrones ou asynchrones. DTD définit les éléments utilisés dans le fichier XML, une liste de leurs attributs et les éléments du niveau suivant.

```
<!ELEMENT labels (label)*>
<!ATTLIST labels _FORMAT CDATA #IMPLIED>
<!ATTLIST labels _JOBNAME CDATA #IMPLIED>
<!ATTLIST labels _QUANTITY CDATA #IMPLIED>
<!ATTLIST labels _PRINTERNAME CDATA #IMPLIED>
<!ELEMENT label (variable)*>
<!ATTLIST label _FORMAT CDATA #IMPLIED>
<!ATTLIST label _JOBNAME CDATA #IMPLIED>
<!ATTLIST label _QUANTITY CDATA #IMPLIED>
```

9.1.5.2. Exemple d'XML Oracle

Voici l'XML Oracle qui fournit les données pour une étiquette (il y a un seul élément **`<label>`**).

```
<?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>
```

Lors de l'exécution de ce fichier XML Oracle, l'étiquette **`serial.lbl`** s'imprimera avec les valeurs suivantes.

Nom de la variable	Valeur de la variable
Article	O Ring
révision	V1
lot	123
Numéro de série	12345
Lot	123
état du numéro de série	Actif
organisation	A1

Il y aura **1** copie imprimée de l'étiquette avec le nom **Serial** dans le spouleur. Le nom d'imprimante n'est pas spécifié dans le fichier XML, l'étiquette s'imprime donc sur l'imprimante définie dans le masque d'étiquette.

9.1.6. Caractéristiques SAP AII XML

NiceLabel Automation peut se présenter comme gestionnaire d'unité RFID, capable d'encoder et d'imprimer des étiquettes RFID. Pour plus d'informations concernant les caractéristiques SAP AII XML, consulter le document **SAP Auto-ID Infrastructure Device Controller Interface** sur les pages Web SAP.

Pour exécuter ce type de fichier de commande, utiliser l'action Exécuter le Fichier de Commande SAP AII XML .

9.1.6.1. Exemple SAP AII XML

Voici l'XML AII SAP qui fournit les données pour une étiquette (il y a un seul élément **<label>**).

```
<?xml version="1.0" encoding="UTF-8"?>
<Command xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Command.xsd">
    <WriteTagData readerID="DEVICE ID">
        <Item>
            <FieldList format="c:\SAP Demo\SAP label.nlbl"
jobName="Writer_Device20040929165746" quantity="1">
            <Field name="EPC">00037000657330</Field>
            <Field name="EPC_TYPE">SGTIN-96</Field>
            <Field name="EPC_URN">urn:autoid:tag:sgtin:3.5.0037000.065774.8</
Field>
            <Field name="PRODUCT">Product</Field>
            <Field name="PRODUCT_DESCRIPTION">Product description</Field>
            </FieldList>
        </Item>
    </WriteTagData>
</Command>
```

Lors de l'exécution de cet exemple SAP AI XML, l'étiquette **c:\SAP Demo\SAP label.nlbl** s'imprime avec les valeurs de variables suivantes.

Nom de la variable	Valeur de la variable
EPC	00037000657330
EPC_TYPE	SGTIN-96
EPC	urn:autoid:tag:sgtin:3.5.0037000.065774.8

PRODUIT	Produit
PRODUCT_DESCRIPTION	Description

Il y aura **1** copie imprimée de l'étiquette avec le nom **Writer_Device2004092916574** dans le spouleur. Le nom de l'imprimante n'est pas spécifié dans le fichier XML, l'étiquette s'imprime donc sur l'imprimante définie dans le masque d'étiquette.

9.1.7. Fichier de commande PAS

Le fichier de commande PAS est un fichier texte qui contient les commandes d'impression Loftware LPS. Les commandes sont exécutées dans l'ordre, de haut en bas. Chaque fichier PAS doit démarrer avec la commande *FORMAT (ouvrir l'étiquette) et se terminer avec la commande *PRINTLABEL.

Le fichier de commande PAS peut s'exécuter à l'aide de l'action Lancer le fichier de commande.

NOTE

Si vous avez migré de Loftware à NiceLabel, vous pouvez utiliser vos fichiers de commande PAS existants avec les contrôles de commande *FORMAT de modèles d'étiquettes NiceLabel (.nlbl) s'il existe un fichier .nlbl avec le même nom que .lwl.

9.1.7.1. Exemple de fichier de commande PAS

```
*FORMAT,\\127.0.0.1\labelcomputer$\labels\ML2193.lwl
*JOBNAME,PastaPallet002
*QUANTITY,1
*PRINTERNUMBER,1
Text0000,New Data
Text0001,New Data
*PRINTLABEL
```

Plus de détails sur les commandes PAS figurent dans le Guide d'utilisateur de Loftware LPS.

9.2. Commandes personnalisées

9.2.1. Utiliser des commandes personnalisées

Les commandes NiceLabel sont utilisées dans les fichiers de commande pour contrôler l'impression des étiquettes. NiceLabel Automation exécute les commandes du fichier de commande, de haut en bas. Pour plus d'informations, voir l'article Spécifications des Fichiers de Commande.

Vous pouvez utiliser la commande personnalisée spécifique, quand elle est disponible en tant qu'actions dans votre produit NiceLabel Automation.

Exemple

Vous pouvez utiliser la commande **SETPRINTPARAM** si vous pouvez sélectionner l'action **Set Print Parameter** (disponible avec les niveaux de produit Pro et Enterprise).

Caractéristiques des commandes de NiceLabel

COMMENTAIRE

```
;
```

Il est conseillé de documenter vos commandes durant le développement du fichier de commande. Cela permettra de décoder ce que le script fait vraiment en jetant un œil dessus de temps en temps. Mettre un point-virgule (`;`) au début de la ligne. Tout ce qui suit le point-virgule sera considéré comme un commentaire et ne sera pas traité.

CLEARVARIABLEVALUES

```
CLEARVARIABLEVALUES
```

Cette commande réinitialise les valeurs des variables à leur valeur par défaut.

CREATEFILE

```
CREATEFILE <nom de fichier> [, <contents>]
```

Cette commande va créer un fichier texte. L'utiliser pour signaler à une application tierce que le processus d'impression a commencé ou s'est terminé, selon l'endroit où est placée la commande. Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées.

DELETEFILE

```
DELETEFILE <nom de fichier>
```

Efface le fichier spécifié. Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées.

EXPORTLABEL

```
EXPORTLABEL ExportFileName [, ExportVariant]
```

La commande permet d'automatiser la commande « Exporter vers l'imprimante » qui se trouve dans l'éditeur d'étiquettes. L'étiquette est exportée directement vers l'imprimante et stockée dans sa mémoire pour une impression hors ligne. L'utilisateur peut rappeler l'étiquette par le clavier de l'imprimante ou en envoyant une commande à l'imprimante. La même fonction est disponible dans l'action Enregistrer l'étiquette dans l'imprimante.

> **NOTE**
> Pour spécifier l'étiquette à exporter, commencer par utiliser la commande **LABEL**.

- **ExportFileName :** Ce paramètre est obligatoire. Il définit le nom du fichier d'exportation. Le contenu du fichier est en langage imprimante.
- **ExportVariant :** Certaines imprimantes supportent plusieurs variantes d'exportation. Durant l'exportation manuelle, l'utilisateur peut choisir la variante d'exportation dans l'interface. Avec la commande EXPORTLABEL, il faut spécifier la variante d'exportation à utiliser. Les variantes se voient dans l'éditeur d'étiquettes, quand le mode d'impression Stocker/Rappeler est activé.
  La première variante de la liste a la valeur 0. La seconde variante a la valeur 1, etc.
  S'il n'y a aucune précision sur le type de variante à utiliser, la valeur 0 est prise par défaut.

Pour plus d'informations sur l'impression hors ligne, voir l'article Utiliser le mode d'impression Stocker/Rappeler.

IGNOREERROR

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

Cette commande spécifie que l'erreur survenant dans le fichier JOB ne stoppera pas le processus d'impression, si l'une des erreurs suivantes se produit :

- Un nom de variable incorrect est utilisé.
- Une valeur incorrecte est envoyée à la variable.
- L'étiquette n'existe pas/n'est pas accessible.
- L'imprimante n'existe pas/n'est pas accessible.

LABEL

```
LABEL <label name> [,<printer_name>]
```

La commande ouvre l'étiquette à imprimer. Si l'étiquette est déjà chargée, elle ne se rouvrira pas. Mettre éventuellement le nom du chemin. Mettre le nom de l'étiquette entre guillemets, si le nom ou le chemin contiennent des espaces. Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées.

L'option **`printer_name`** spécifie l'imprimante pour laquelle l'étiquette sera ouverte. Utiliser ce paramètre pour remplacer l'imprimante qui est enregistrée dans le masque de l'étiquette. Si le pilote de cette imprimante n'est pas installé ou pas disponible, la commande signalera une erreur.

MESSAGEBOX

```
MESSAGEBOX <message> [,<caption>]
```

Cette commande enregistre le **`message`** personnalisé dans le journal du déclencheur. Si le message contient des caractères d'espacement ou des virgules, il faut placer le texte entre guillemets (**"**).

PORT

```
PORT <file name> [, APPEND]
```

Cette commande remplace le port défini dans le pilote d'imprimante et redirige l'impression vers un fichier. Si le nom de chemin ou de fichier contient des espaces, mettre la valeur entre guillemets (**"**). Utiliser la syntaxe UNC pour les ressources réseau. Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées.

Le paramètre **APPEND** est facultatif. Par défaut, le fichier est écrasé. Utiliser ce paramètre pour joindre des données à un fichier existant.

À chaque utilisation d'une commande PORT dans le fichier JOB, elle reste valable jusqu'à la commande PORT suivante, ou jusqu'à la fin du fichier (ce qui arrive en premier). Si une commande PRINTER suit l'exécution de la commande PORT, le paramètre PORT va écraser le port défini pour l'imprimante sélectionnée. Pour utiliser le port défini pour l'imprimante sélectionnée, il faut utiliser une autre commande PORT avec une valeur vide, telle que **`PORT = ""`**.

PRINT

```
PRINT <quantité> [,<skip> [,<identical label copies> [,number of label sets]]]
- Imprimer, quantité, saut, copies identiques, nombre de lots d'étiquettes
```

Cette commande lance le processus d'impression.

- **Quantité :** Spécifie le nombre d'étiquettes imprimées.
  - **<number> :** Le nombre d'étiquettes spécifié est imprimé.
  - **VARIABLE :** Spécifie qu'une variable d'étiquette est définie comme *quantité variable* et contient le nombre d'étiquettes à imprimer. L'étiquette détermine le nombre d'étiquettes à imprimer.
  - **IILIMITÉ :** Avec une base de données comportant les valeurs des objets, l'impression illimitée imprimera autant d'étiquettes qu'il y a d'enregistrements dans la base de données. Sans base de données, le nombre maximal d'étiquettes supporté par l'imprimante thermique sera imprimé.
- **Sauter :** Spécifie le nombre d'étiquettes à sauter sur la première page. Ce paramètre est utilisé pour l'impression d'étiquettes sur des feuilles de papier. Quand une partie de la page a déjà été utilisée, réutiliser la même feuille en déplaçant le point de départ de la première étiquette.
- **Le paramètre « identical label copies » (copies d'étiquettes identiques) :** Spécifie le nombre de copies de la même étiquette à imprimer.

- **Nombre de jeux d'étiquettes.** Spécifie combien de fois le processus d'impression complet doit être répété.

> **NOTE**
> Il faut que les valeurs de quantité soient saisies en valeurs numériques, et pas en chaînes de caractères. Ne pas placer les valeurs entre guillemets.

PRINTER

```
PRINTER <nom d'imprimante>
```

Cette commande remplace l'imprimante définie dans le fichier d'étiquette. Si le nom d'imprimante contient des caractères espace, il faut le mettre entre guillemets(**"**).

Utiliser le nom d'imprimante tel qu'il est affiché dans l'éditeur d'étiquettes. Les imprimantes ont généralement les mêmes noms que celles du panneau de configuration Imprimantes et Fax, mais pas toujours. Avec des imprimantes réseau, le nom peut s'afficher avec la syntaxe **`\\server\share`**.

PRINTJOBNAME

```
PRINTJOBNAME
```

Cette commande spécifie le nom du travail d'impression visible dans le Spouleur Windows. Si le nom contient des caractères d'espacement ou virgules, il faut les mettre entre guillemets (**"**).

SESSIONEND

```
SESSIONEND
```

Cette commande ferme le flux d'impression. Voir aussi **SESSIONSTART** (début de session).

> **NOTE**
> `SESSIONEND` doit être le seul élément envoyé dans l'action Envoyer une commande personnalisée. Si vous souhaitez envoyer d'autres commandes, utiliser différentes actions Envoyer une commande personnalisée.

SESSIONPRINT

```
SESSIONPRINT <quantity> [,<skip>]
```

Cette commande imprime l'étiquette actuellement référencée et l'ajoute à la session d'impression actuellement ouverte. Il est possible d'utiliser plusieurs commandes SESSIONPRINT l'une après l'autre et

de joindre les étiquettes référencées dans un flux d'impression unique. Le flux ne se fermera pas avant la commande SESSIONEND. Les paramètres quantity (quantité) et skip (sauter) ont la même signification que dans la commande PRINT. Voir aussi la commande **SESSIONSTART** (début de session).

- **Quantité :** Spécifie le nombre d'étiquettes à imprimer.
- **Sauter :** Spécifie le nombre d'étiquettes à sauter sur la première page. Ce paramètre est utilisé pour l'impression d'étiquettes sur des feuilles de papier. Quand une partie de la page a déjà été utilisée, réutiliser la même feuille en déplaçant le point de départ de la première étiquette.

SESSIONSTART

```
SESSIONSTART
```

Cette commande initie le type d'impression session-print.

Les trois commandes d'impression de session (**SESSIONSTART**, **SESSIONPRINT** et **SESSIONEND**) sont utilisées ensemble. Avec la commande PRINT, toutes les données d'étiquettes sont envoyées à l'imprimante dans un travail d'impression différent. Pour joindre les données d'étiquettes dans un flux d'impression, utiliser les commandes d'impression de session. Pour ce faire, commencer avec la commande SESSIONSTART suivie de n'importe quel nombre de commandes SESSIONPRINT. La séquence se termine par la commande SESSIONEND.

Utiliser ces commandes pour optimiser le processus d'impression des étiquettes. L'impression d'étiquettes appartenant à un même travail d'impression est beaucoup plus rapide que d'imprimer des étiquettes appartenant à plusieurs travaux d'impression.

Utiliser les règles ci-dessous pour vous assurer la session d'impression ne soit pas rompue :

- Ne pas changer d'étiquette pendant une session.
- Ne pas changer d'imprimante pendant une session.
- Il faut paramétrer des valeurs pour toutes les variables de l'étiquette durant la session, même si certaines variables ont une valeur vide.

SET

```
SET <nom>=<valeur> [,<pas> [,<nombre ou répétitions>]]
```

Cette commande assigne le **`nom`** de variable à la **`valeur`**. La variable doit être définie sur l'étiquette, sinon une erreur sera signalée. Une erreur surviendra si la variable n'est pas sur l'étiquette. L'**`incrément`** et le **`nombre de répétitions`** sont des paramètres pour les variables de comptage. Ces paramètres spécifient l'incrémentation du compteur et le nombre d'étiquettes avant que le compteur change de valeur.

Si la valeur contient des caractères d'espacement ou virgule, les mettre entre guillemets (**"**). Voir aussi **TEXTQUALIFIER**.

Pour assigner une valeur en plusieurs lignes, utiliser **`\r\n`** pour encoder un caractère de retour à la ligne. **`\r`** est remplacé par CR (Retour Chariot) et **`\n`** est remplacé par LF (Nouvelle Ligne).

Attention en paramétrant les valeurs de variables procurant des données pour les images des étiquettes, puisque la barre oblique inversée peut être remplacée par un autre caractère.

Exemple

Avec la valeur "c:\Mes Images\raw.jpg" pour la variable, le "\r" sera remplacé par le caractère CR.

SETPRINTPARAM

```
SETPRINTPARAM <nom paramètre> = <valeur>
```

Cette commande permet d'affiner les paramètres de l'imprimante avant d'imprimer. Les paramètres supportés pour les réglages d'imprimante (**`paramname`**) sont :

- **PAPERBIN :** Spécifie le réservoir contenant le support d'étiquette. Si l'imprimante est équipée de plus d'un bac à papier/bac à étiquettes, contrôler celui qui sera utilisé pour l'impression. Le nom du bac à papier doit provenir du pilote d'imprimante.
- **PRINTSPEED :** Spécifie la vitesse d'impression. Les valeurs acceptables varient d'une imprimante à l'autre. Consulter le manuel de l'imprimante pour connaître les valeurs exactes.
- **PRINTDARKNESS :** Spécifie le contraste de l'impression. Les valeurs acceptables varient d'une imprimante à l'autre. Consulter le manuel de l'imprimante pour connaître les valeurs exactes.
- **PRINTOFFSETX :** Spécifie la marge de gauche pour tous les objets imprimables. La valeur du paramètre est numérique, exprimée en pixels. Elle peut être positive ou négative.
- **PRINTOFFSETY :** Spécifie la marge supérieure pour tous les objets imprimables. La valeur du paramètre est numérique, exprimée en pixels. Elle peut être positive ou négative.
- **PRINTERSETTINGS :** Spécifie les paramètres personnalisés à appliquer au travail d'impression. Le paramètre a besoin du DEVMODE entier de l'imprimante ciblée, il est fourni par une chaîne de caractères codée en Base64. Le DEVMODE contient tous les paramètres du pilote d'imprimante (vitesse, contraste, décalage et autre). Pour plus d'informations, voir l'article Comprendre les paramètres d'imprimante et DEVMODE.

  NOTE
  La chaîne de caractères codée en Base64 doit être fournie entre guillemets (").

TEXTQUALIFIER

```
TEXTQUALIFIER <caractère>
```

Le délimiteur de texte est le caractère qui entoure la valeur de la donnée qui est assignée à une variable. Si la valeur comprend des caractères d'espacement, ils doivent être inclus dans des délimiteurs de texte. Le délimiteur de texte par défaut sont les guillemets (**"**). Comme les guillemets sont utilisés comme raccourci pour les unités de mesure en pouces, il est parfois difficile de passer les données avec le signe pouce dans les fichiers JOB. Il faut alors utiliser des guillemets doubles pour en encoder un ou utiliser TEXTQUALIFIER.

Exemple

```
TEXTQUALIFIER%

SET Variable = %EPAK 12"X10 7/32"%
```

9.3. Accès aux ressources réseau partagées

Cet article définit les recommandations à suivre lors de l'utilisation de ressources de réseau partagées.

- **Droits d'utilisateur pour le mode de service.** NiceLabel Automation s'exécute en mode service sous le compte utilisateur spécifié et hérite des droits d'accès de ce compte. Pour que NiceLabel Automation puisse ouvrir les fichiers d'étiquettes et utiliser les pilotes d'imprimante, le compte utilisateur associé doit disposer des mêmes privilèges. Pour plus d'informations, consulter l'article Fonctionnement en mode service .
- **Notation UNC pour les partages réseau.** Pour accéder à un fichier sur un disque réseau, utiliser toujours la syntaxe UNC (Convention Universelle de Noms) et pas les lettres des disques mappés. UNC est une convention d'appellation de chemins qui spécifie et mappe les disques réseau. NiceLabel Automation tente automatiquement de remplacer la syntaxe avec le lien sur le disque par la syntaxe UNC.

> **Exemple**
>
> Si l'adresse du fichier est **`G:\Labels\label.nlbl`**, la notation UNC sera **`\\server\share\Labels\label.nlbl`** (où le disque G : est lié à **`\\server\share`**).

- **Notation pour accéder aux fichiers dans Control Center.** Pour ouvrir le fichier du Stockage de documents dans Control Center, utiliser la notation HTTP telle que **`http://servername:8080/label.lbl`** ou la notation WebDAV telle que **`\\servername@8080\DavWWWRoot\label.lbl`**. Notes additionnelles :
  - Le compte utilisateur avec lequel le service NiceLabel Automation est exécuté est également utilisé pour obtenir des fichiers à partir du Stockage de documents. Cet utilisateur doit être configuré dans la configuration de Control Center. Il aura ainsi accès aux fichiers du Stockage de documents.
  - L'accès WebDAV n'est utilisable qu'avec l'authentification d'utilisateur Windows dans le Control Center.

> **NOTE**
> Le Stockage de documents est disponible avec les produits **LMS Enterprise** et **LMS Pro** .

- **Disponibilité des pilotes d'imprimantes.** Pour imprimer des étiquettes sur une imprimante partagée en réseau, le pilote de l'imprimante doit être accessible depuis le serveur sur lequel NiceLabel Automation est installé. Vérifier que le compte utilisateur sous lequel NiceLabel Automation s'exécute a les droits d'accès au pilote d'imprimante. Si l'imprimante en réseau vient d'être installée sur la machine, il est possible que NiceLabel Automation ne la voit pas avant d'avoir redémarré le Service. Pour permettre la notification automatique de nouveaux pilotes d'imprimantes réseau, il faut activer la règle d'entrée correspondante dans le pare-feu Windows. Pour plus d'informations, consulter la Base de Connaissances article.

9.4. Stockage de documents et contrôle des versions des fichiers de configuration

Le Stockage de documents est une fonctionnalité de NiceLabel Control Center. Elle permet de faire fonctionner NiceLabel Control Center comme un référentiel de fichiers partagé sur le serveur, dans lequel les utilisateurs peuvent stocker et récupérer leurs fichiers et contrôler les différentes révisions.

L'onglet contextuel **Stockage de documents** permet d'effectuer des actions de stockage de documents directement depuis Automation Builder. Il n'est donc plus nécessaire d'accéder au fichier Automation pour l'ouvrir dans NiceLabel Control Center.

> **NOTE**
> Cet onglet contextuel a besoin d'une connexion avec NiceLabel Control Center. Ce type de configuration exige une licence LMS Enterprise.

Le groupe **Révision** permet d'effectuer les actions disponibles pour le Stockage de documents :

- **Extraire (Check Out) :** Extrait le fichier du Stockage de documents de NiceLabel Control Center pour qu'il puisse être modifié. Le fichier extrait est alors marqué et verrouillé pour tous les autres utilisateurs. Tous les autres utilisateurs verront la version actuelle du fichier, tandis que l'auteur (éditeur) peut déjà travailler sur un nouveau brouillon.

> **NOTE**
> Après ouverture d'un document du Stockage de documents, (**Fichier** > **Ouvrir** > **Stockage de documents**), les commandes d'édition restent désactivées jusqu'à ce que le document soit extrait.

- **Archiver (Check In) :** Enregistre le fichier dans le Stockage de documents de NiceLabel Control Center après l'avoir modifié. Lors de l'archivage du fichier, le nombre de révisions du fichier s'incrémente de un. Le commentaire saisi est inscrit dans le journal du fichier.
- **Annuler l'extraction :** Annule l'extraction du fichier en cours et redonne aux autres utilisateurs l'accès à ce fichier.

  AVERTISSEMENT
  En cliquant sur **Annuler l'extraction**, on perd toutes les modifications effectuées depuis l'extraction du fichier.

- **Stockage de documents :** Ouvre le stockage de documents de l'NiceLabel Control Center connecté.

9.5. Accéder aux bases de données

Chaque fois que NiceLabel Automation doit accéder aux données d'une base de données, vérifier que le pilote de base de donnée requis est installé dans le système Windows. Les pilotes de bases de données sont fournis par la société qui a développé le logiciel de la base de données. Le pilote installé doit correspondre au nombre de bits de votre système Windows. Le logiciel NiceLabel tourne toujours sous le nombre de bits du système Windows.

9.5.1. Windows 32 bits

Avec Windows 32 bits, installer uniquement des pilotes de base de données 32 bits. Le même pilote de base de données est utilisé pour configurer le déclencheur dans Automation Builder et effectuer l'exécution du déclencheur dans le Service NiceLabel Automation. Tous les composants NiceLabel Automation sont exécutés comme des applications 32 bits.

9.5.2. Windows 64 bits

Avec Windows 64 bits, installer des pilotes de base de données 32 bits ou 64 bits. Les applications exécutées en 64 bits utilisent les pilotes de base de données 64 bits. Les applications exécutées en 32 bits utilisent les pilotes de base de données 32 bits.

Par défaut, le service Automation tourne en 64 bits. Il utilise donc les pilotes de base de données 64 bits pour se connecter aux bases de données. S'il n'y a pas de pilote de base de données 64 bits sur le système où tourne le service Automation, la connexion à la base de données passe sur le **NiceLabel Proxy** qui tourne toujours en 32 bits. Cette application s'exécute toujours comme une application 32 bits.

9.6. Remplacement automatique de la police

Il est possible de créer des masques d'étiquettes pour imprimer des objets texte formatés en polices imprimante. Voici les polices qui sont stockées dans la mémoire de votre imprimante. Si vous essayez

d'imprimer de telles étiquettes sur un autre type d'imprimante, les polices internes sélectionnées ne seront peut-être pas disponibles. La nouvelle imprimante supporte probablement un jeu de polices internes complètement différent. Les polices peuvent être semblables dans ce cas, mais disponibles sous un autre nom.

Une différence de polices peut également se produire si la police Truetype que vous utilisez sur vos étiquettes n'est pas installée sur l'ordinateur sur lequel vous exécutez Designer pour créer et imprimer vos étiquettes.

Designer peut être configuré pour remplacer automatiquement les polices utilisées dans l'étiquette par des polices compatibles. Dans ce cas, Designer mappe et remplace les polices en utilisant leurs noms. Si la police originale n'est pas disponible, Designer essaiera d'utiliser la première police de remplacement disponible, comme définie dans la table de mappage.

S'il n'y a pas de polices de remplacement appropriées, Designer utilise la police Arial Truetype.

NOTE

Après avoir configuré la police de remplacement, les règles du mappage s'exécutent lorsque vous changez d'imprimante pour votre étiquetage.

Configuration du mappage de police

1. Ouvrir l'explorateur de fichiers et rechercher le dossier suivant :

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10
```

2. Copier le fichier **fontmapping.def** file dans **fontmapping.local.def**.
3. Ouvrir le fichier **fontmapping.local.def** dans un éditeur de texte XML.
4. Dans l'élément **FontMappings**, créer un nouvel élément avec un nom personnalisé.
5. Dans le nouvel élément, créer au moins deux éléments portant le nom **Mappage**.
   - La valeur du premier élément portant le nom Mappage doit contenir le nom de la police d'origine.
   - La valeur du second élément portant le nom Mappage doit contenir le nom de la police de remplacement.

> **NOTE**
> Il est autorisé de mettre d'autres éléments de mappage avec de nouveaux noms de police. Si la première police de remplacement n'est pas disponible, Designer essaiera la suivante. Si aucune police de remplacement n'est disponible, Arial Truetype sera utilisée.

> **NOTE**
> Le fichier **fontmapping.local.def** est le vôtre. Il est conservé lors des mises à niveau. D'un autre côté, **fontmapping.def** appartient à NiceLabel et est écrasé lors des mises à jour. Ne pas modifier le fichier **fontmapping.def**.

Exemple de configuration de mappage

Dans l'exemple ci-dessous, deux mappages sont définis.

- Le premier mappage va convertir toute police **Avery** en police correspondante **Novexx**. Par exemple, la police **Avery YT100** sera remplacée par **Novexx YT100**, la police **Avery 1** sera remplacée par **Novexx 1**. Si la police Novexx n'est pas disponible, **Arial** Truetype sera utilisé.
- Le deuxième mappage va convertir **Avery YT100** en **Novexx YT104**. Si cette police n'est pas disponible, la police **Zebra 0** sera utilisée. Et si cette police n'est pas disponible, **Arial** Truetype sera utilisée.
- Le second mappage écrasera le premier.

```
<?xml version="1.0" encoding="utf-8"?>
<FontMappings>
    <AveryNovexx>
        <Mapping>Avery</Mapping>
        <Mapping>Novexx</Mapping>
    </AveryNovexx>
    <TextReplacement>
        <Mapping>Avery YT100</Mapping>
        <Mapping>Novexx YT104</Mapping>
        <Mapping>Zebra 0</Mapping>
    </TextReplacement>
</FontMappings>
```

9.7. Automatisation des rapports

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise**, **LMS Pro**, **Label Cloud Business** et **Label Cloud Compliance**.

Combinez les données système de votre entreprise avec des étiquettes contenant des **Rapports**. Créez des rapports dans NiceLabel Designer et remplissez vos rapports de données provenant des systèmes de votre entreprise dans NiceLabel Automation.

Vos rapports terminés utilisent les données système de votre entreprise pour l'impression. NiceLabel Automation :

- **Reçoit** les **données** système de votre entreprise.
- **Analyse vos données** avec un **filtre de données**.

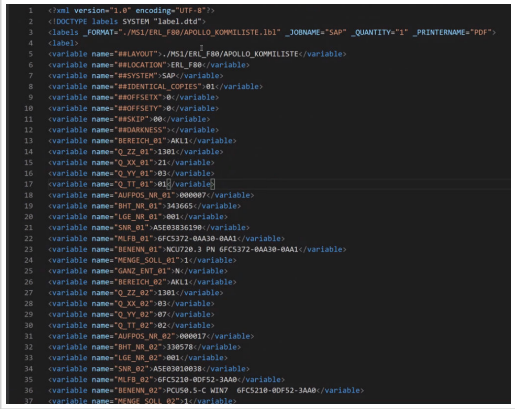
- **Remplit votre rapport** de données analysées.
- **Exécute l'impression** pour vos nouveaux rapports remplis de données avec un **déclencheur**.

9.7.1. Création de bases de données temporaires

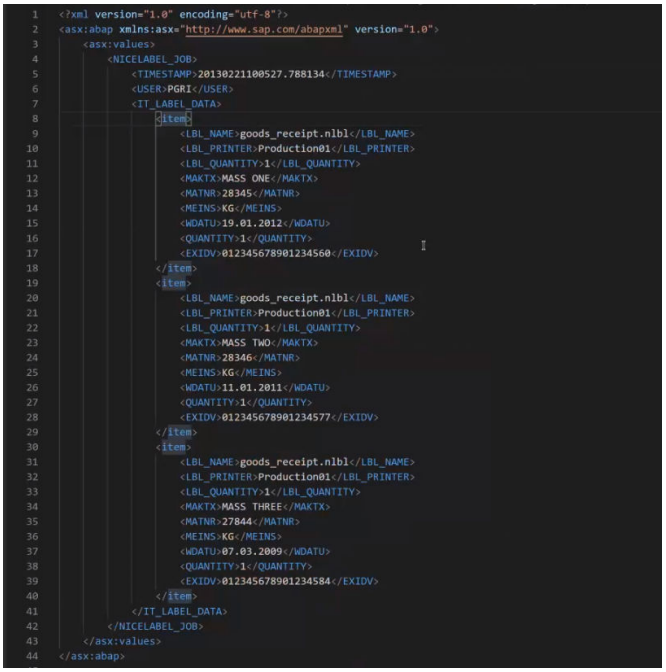
Vous devez élaborer vos **rapports** avant de les automatiser. Pour élaborer des rapports, vous devez les connecter à une **base de données**.

Si vous utilisez des données provenant de systèmes d'entreprises externes pour les rapports, vous n'avez **aucune base de données** avec laquelle travailler. Pour fonctionner de façon efficace, les données de rapport ont besoin d'une **structure hiérarchique** avec des éléments clairement définis.

Créez une **base de données temporaire** avec les données provenant du système de votre entreprise (généralement des données au format .XML ou .JSON). Avec votre base de données temporaire, élaborez et configurez vos rapports. **Votre base de données temporaire sert uniquement à des fins de configuration**. Les rapports configurés sont imprimés avec des déclencheurs Automation, pas votre base de données temporaire.



Données provenant du système de votre entreprise sans hiérarchie.



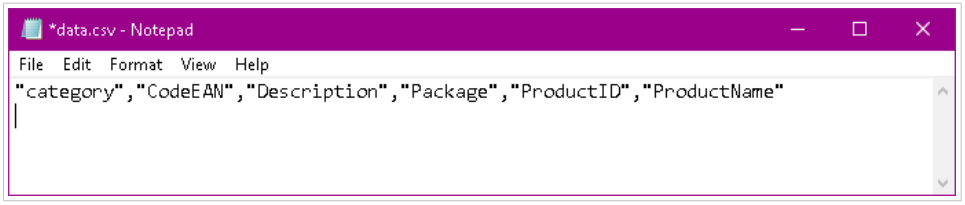
Les données avec une hiérarchie que vous pouvez utiliser pour imprimer des rapports.

```
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="Pasta.xsd" generated="2020-08-31T15:43:51">
  <Total>211</Total>
  <Pasta>
    <ProductID>PAS501</ProductID>
    <CodeEAN>8021228310001</CodeEAN>
    <ProductName>BIGOLI 250G</ProductName>
    <Package>6</Package>
    <category>Long pasta</category>
    <Description>A long, thin, cylindrical pasta of Italian origin. Spaghetti is made of semolina or flour and
    water.</Description>
  </Pasta>
  <Pasta>
    <ProductID>PAS502GI</ProductID>
    <CodeEAN>8021228310018</CodeEAN>
    <ProductName>TAGLIATELLE 250G</ProductName>
    <Package>6</Package>
    <category>Ribbon-cut pasta</category>
    <Description>Ribbon style pasta are often rolled flat and then cut. This can be done by hand or
    mechanically.</Description>
  </Pasta>
```


Échantillon de données XML dont vous avez besoin pour élaborer des rapports (non analysés).

Pour votre base de données temporaire, créez un fichier de base de données texte au format .CSV. Vous avez plusieurs options :

1. Convertissez manuellement votre structure de données en fichier texte .CSV :



Création manuelle de base de données texte temporaires au format .CSV.

2. Utilisez un outil de conversion courant XML/Excel ou XML/CSV et formatez manuellement vos données dans une feuille de calculs.

3. Utilisez un **filtre de données** Automation pour analyser automatiquement vos données dans une base de données texte au format .CSV (consultez votre échantillon de fichier **XML ou CSV** inclus pour des informations complémentaires).

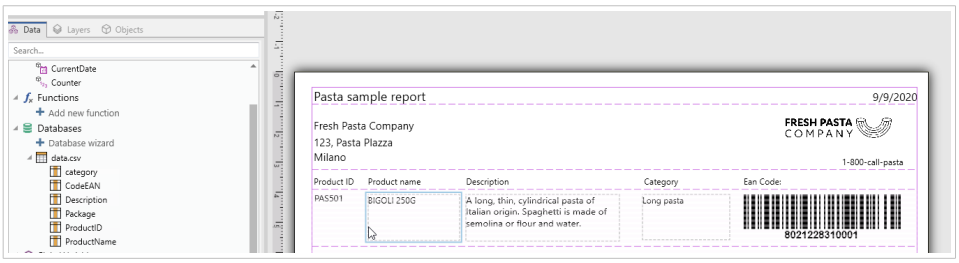
9.7.2. Élaboration de rapports automatisés

Ouvrez **Designer** pour créer votre **rapport** avec votre base de données texte temporaire au format .CSV :

1. **Connectez** votre base de données texte temporaire au format .CSV à votre **rapport**.

2. **Élaborez** votre rapport à l'aide de vos données texte CSV en tant que variables pour les objets dans votre **Définition du répétiteur**.

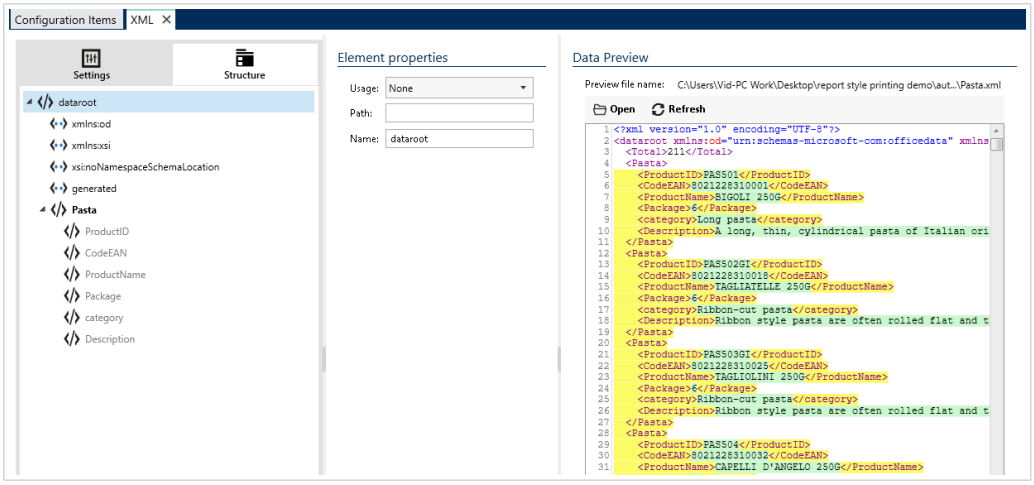
> **NOTE**
> **Associez** vos **noms de variables** aux **noms de données** dans votre filtre de données pour cartographier vos données de façon appropriée.



Élaboration de rapports avec votre base de données texte temporaire au format .CSV.

9.7.3. Création de filtres de données

Créez un **filtre de données** dans Automation pour l'utiliser dans votre rapport (consultez votre exemple de déclencheur inclus pour des informations complémentaires sur la création de filtres XML) :



Configuration de votre filtre de données XML.

Conseils sur la création de filtres de données pour les rapports :

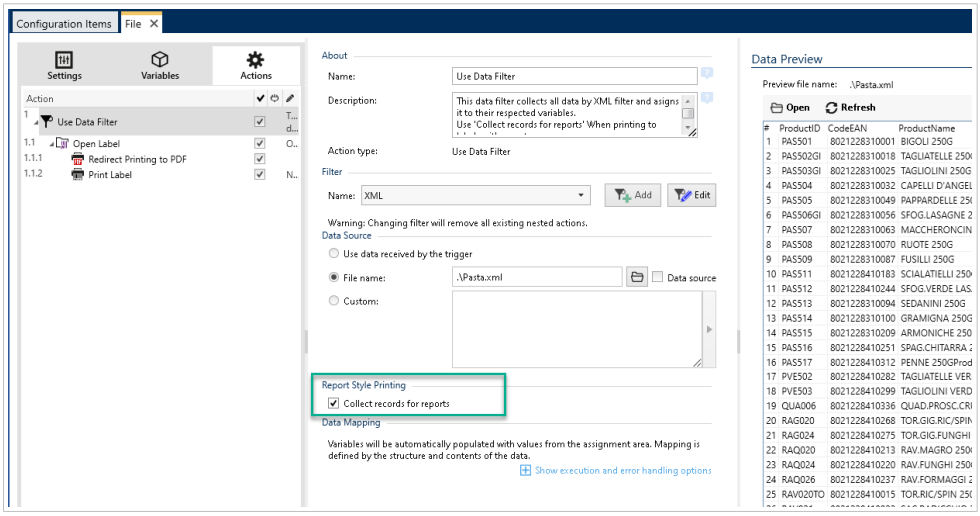
- Vous pouvez utiliser l'**Assistant d'importation de structure** pour importer votre structure de données.
- Nommez vos éléments dans les blocs de données d'éléments répétables.
- Définissez vos éléments en tant que zones d'affectation ou les sous-éléments en tant que variables.

9.7.4. Création de déclencheurs pour votre nouveau filtre de données

NOTE

Pour que votre déclencheur fonctionne, les noms de vos sources de données dans Designer doivent correspondre aux noms de champs de blocs de données dans votre filtre de données.

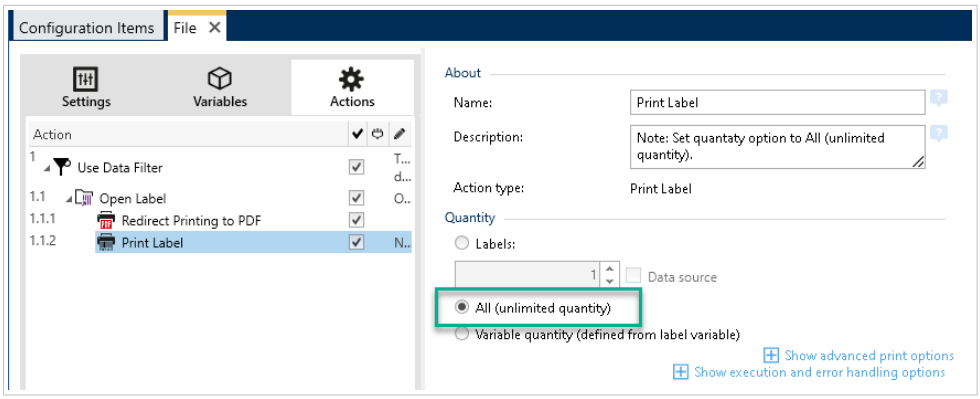
- Incluez une action **Utiliser un filtre de données** pour appliquer votre filtre de données.



- Activez **Collecter des dossiers pour le rapport** afin de collecter tous les enregistrements dans un tableau qu'Automation utilise pour ouvrir les étiquettes de votre rapport.

> **NOTE**
> Utilisez les mêmes noms de structures de données dans vos rapports, déclencheurs et filtres.

- Incluez une action **Imprimer l'étiquette**.



- Activez **Tous (quantité illimitée)**.

Exécutez votre déclencheur. Votre rapport s'actualise alors automatiquement avec les nouvelles données et s'imprime avec votre déclencheur.

9.8. Changer les paramètres d'impression multi threads par défaut

INFO NIVEAU DE PRODUIT

Cette fonctionnalité est disponible dans **LMS Enterprise**, **LMS Pro**, **Label Cloud Business** et **Label Cloud Compliance**.

Chaque produit NiceLabel Automation peut tirer parti de multiples cœurs de processeur, chaque cœur exécute un processus d'impression indépendamment. La moitié des cœurs est utilisée pour exécuter des tâches simultanées *normales*, et l'autre moitié pour le traitement des tâches simultanées *session-impression* .

NOTE

En temps normal, il ne faut jamais changer les paramètres par défaut. Ils ne peuvent être modifiés qu'en connaissance de cause.

Pour changer le nombre de threads concurrentes d'impression, effectuer les opérations suivantes :

1. Ouvrir le fichier **`product.config`** dans un éditeur de texte.
   Le fichier est ici :

   ```
   %PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
   ```

2. Changer les valeurs des éléments **MaxConcurrentPrintProcesses** et **MaxConcurrentSessionPrintProcesses**.

   ```
   <configuration>
       <IntegrationService>
           <MaxConcurrentPrintProcesses>1</MaxConcurrentPrintProcesses>
           <MaxConcurrentSessionPrintProcesses>1</
   MaxConcurrentSessionPrintProcesses>
       </IntegrationService>
   </configuration>
   ```

3. Enregistrer le fichier. NiceLabel Automation met automatiquement à jour le service avec le nouveau nombre de threads d'impression.

9.9. Compatibilité avec les produits NiceWatch

NiceLabel Automation vous permet de charger des configurations de déclencheurs conçues en utilisant les produits NiceWatch. Dans la majorité des cas, une configuration de NiceWatch peut être exécutée dans NiceLabel Automation sans aucune modification.

Les produits NiceLabel Automation utilisent les nouveaux moteurs d'impression en .NET, optimisés en performances avec un faible encombrement mémoire. Le nouveau moteur d'impression ne supporte pas toutes les options de création des étiquettes disponibles dans l'éditeur d'étiquettes. Chaque nouvelle mouture de NiceLabel Automation en diminue le nombre mais certaines fonctionnalités restent indisponibles.

Résoudre les problèmes d'incompatibilité

NiceLabel Automation émet une alerte à chaque tentative d'impression d'étiquettes existantes qui contiennent des fonctionnalités indisponibles dans le nouveau moteur d'impression.

S'il y a des incompatibilités entre les fichiers de configuration ou les masques d'étiquettes NiceWatch, Automation vous avertit concernant :

- **Compatibilité avec la configuration de déclencheur :** À l'ouverture de la configuration de NiceWatch (fichier .MIS), NiceLabel Automation la compare aux éléments supportés. Tous les fonctionnalités de NiceWatch ne sont pas disponibles dans NiceLabel Automation. Certaines sont totalement indisponibles, d'autres sont configurées différemment. Si le fichier .MIS contient des fonctionnalités non supportés, elles seront listées. Automation supprime ces fonctionnalités de la configuration.
  Dans ce cas, il faudra ouvrir le fichier .MIS dans Automation Builder et résoudre les problèmes d'incompatibilité. Il faudra utiliser la fonctionnalité disponible de NiceLabel Automation pour recréer la configuration.
- **Compatibilité avec les masques d'étiquettes :** Si vos masques d'étiquettes existants contiennent des fonctionnalités de moteur d'impression non supportées telles que fournies par NiceLabel Automation, vous verrez des messages d'erreurs dans le panneau **Journal**. Cette information est visible dans Automation Builder (durant la création des déclencheurs) ou dans Automation Manager (lors de l'exécution des déclencheurs).
  Dans ce cas, il faut ouvrir le fichier de l'étiquette dans l'éditeur d'étiquettes et enlever de l'étiquette les éléments non supportés.

NOTE

Pour plus d'informations concernant les incompatibilités avec NiceWatch et l'éditeur d'étiquettes, consulter laNiceLabel Automation Compatibility with NiceLabel Designers V6 and NiceWatch V5.

Ouvrir la configuration NiceWatch pour l'éditer

Ouvrir la configuration NiceWatch existante (fichier .MIS) dans Automation Builder et l'éditer dans Automation Builder. Vous pouvez enregistrer la configuration en tant que fichier .MISX uniquement.

Pour éditer la configuration NiceWatch, effectuer les opérations suivantes :

1. Démarrer Automation Builder.
2. Sélectionner **Fichier** > **Ouvrir fichier NiceWatch**.
3. Dans la boîte de dialogue **Ouvrir**, rechercher le fichier de configuration NiceWatch (fichier .MIS).
4. Cliquer sur **OK**.
5. Si la configuration contient des fonctionnalités non supportées, elles sont affichées sous forme de liste. Automation les supprime de la configuration.

Ouvrir la configuration NiceWatch pour l'exécuter

Il est possible d'ouvrir la configuration NiceWatch (fichier .MIS) dans Automation Manager sans conversion au format de fichier NiceLabel Automation (fichier .MISX). Si les déclencheurs de NiceWatch sont compatibles avec NiceLabel Automation, ils sont directement utilisables.

Pour éditer et déployer la configuration NiceWatch, effectuer les opérations suivantes :

1. Démarrer Automation Manager.
2. Cliquer sur le bouton **+ Ajouter**.
3. Dans la boîte de dialogue **Ouvrir**, changer le type de fichier en **Configuration NiceWatch**.
4. Rechercher le fichier de configuration NiceWatch (fichier .MIS).
5. Cliquer sur **OK**.
6. Le déclencheur de la configuration sélectionnée sera affiché dans Automation Manager. Pour lancer le déclencheur, le sélectionner et cliquer sur **Démarrer**.

NOTE

S'il y a un problème de compatibilité avec la configuration NiceWatch, il faudra l'ouvrir dans Automation Builder et le reconfigurer.

9.10. Contrôler le Service Automation avec les paramètres de ligne de commande

Lire cet article pour apprendre à utiliser l'invite de commande pour :

- Démarrer ou arrêter les services Automation.
- Contrôler les configurations à charger.
- Contrôler les déclencheurs à activer.

> **NOTE**
> Il faut lancer **Invite de Commande** en mode élevé (avec les permissions d'administrateur). Cliquer à droite sur `cmd.exe` et sélectionner **Exécuter en tant qu'administrateur**.

Démarrer et arrêter les services

Pour démarrer les deux services, utiliser les commandes suivantes :

```
net start NiceLabelProxyService2019

net start NiceLabelAutomationService2019
```

Pour ouvrir le fichier de configuration quand le Service est démarré, utiliser :

```
net start NiceLabelAutomationService2019 [Configuration]
```

Par exemple :

```
net start NiceLabelAutomationService2019 "c:\Project\configuration.MISX"
```

Pour arrêter les services, utiliser les commandes suivantes :

```
net stop NiceLabelProxyService2019

net stop NiceLabelAutomationService2019
```

Gérer les configurations et déclencheurs

Le service de NiceLabel Automation peut contrôler les paramètres de la ligne de commande en utilisant Automation Manager. Utiliser la syntaxe générale suivante :

```
NiceLabelAutomationManager.exe COMMAND Configuration [NomDéclencheur] [/SHOWUI]
```

> **NOTE**
> Inclure le chemin entier dans le nom de la configuration. Ne pas utiliser que le nom de fichier.

Pour ajouter (ADD) une configuration

La configuration fournie sera chargée dans le service. Aucun déclencheur ne démarre. Avec le paramètre `/SHOWUI` inclus, Automation Managerdémarrera.

```
NiceLabelAutomationManager.exe ADD c:\Project\configuration.MISX /SHOWUI
```

Pour recharger (RELOAD) la configuration

La configuration fournie sera rechargée dans le service. L'état d'exécution de tous les déclencheurs sera conservé. Recharger la configuration force la mise à jour de tous les fichiers en mémoire cache pour cette configuration. Pour plus d'informations, consulter l'article Mise en cache de fichiers. Avec le paramètre `/SHOWUI` inclus, Automation Managerdémarrera.

```
NiceLabelAutomationManager.exe RELOAD c:\Project\configuration.MISX /SHOWUI
```

Pour supprimer (REMOVE) une configuration

La configuration fournie et tous ses déclencheurs seront déchargés du service.

```
NiceLabelAutomationManager.exe REMOVE c:\Project\configuration.MISX
```

Pour démarrer (START) un déclencheur

Le déclencheur référencé démarrera dans la configuration déjà chargée.

```
NiceLabelAutomationManager.exe START c:\Project\configuration.MISX CSVTrigger
```

Pour arrêter (STOP) un déclencheur

Le déclencheur référencé s'arrêtera dans la configuration déjà chargée.

```
NiceLabelAutomationManager.exe STOP c:\Project\configuration.MISX CSVTrigger
```

Codes d'états
Les codes d'états fournissent un retour d'informations concernant l'exécution de la ligne de commande. Pour activer le retour des codes d'états, utiliser la syntaxe de ligne de commande suivante :

```
start /wait NiceLabelAutomationManager.exe COMMAND Configuration [TriggerName]
[/SHOWUI]
```

Les codes d'états sont capturés dans la variable du système **`errorlevel`**. Pour voir le code d'état, exécuter la commande suivante :

```
echo %errorlevel%
```

Liste de codes d'états :

Code d'état	Description
0	Aucune erreur survenue
100	Nom du fichier de configuration introuvable
101	La configuration ne peut pas être chargée
200	Déclencheur introuvable
201	Le déclencheur ne peut pas démarrer

Fourniture des informations d'identification de l'utilisateur pour l'authentification de l'application
Si le système LMS Enterprise ou LMS Pro est configuré pour utiliser l'**Authentification de l'application** (et non l'**Authentification Windows**) il faut donner des identifiants utilisateur avec suffisamment de droits pour gérer les configurations et les déclencheurs.

Il y a deux paramètres en ligne de commande utilisables :

- -USER : **`[username]`**. Dans lequel **`[username]`** est réservé au nom de l'utilisateur actuel.
- -PASSWORD : **`[password]`**. Dans lequel **`[password]`** est réservé au mot de passe actuel.

9.11. Remplacement de la chaîne de connexion à la base de données

Un fichier de configuration du Service Automation peut comporter des commandes de remplacement de la chaîne de connexion à la base de données.

Vous pouvez configurer le service pour remplacer certaines parties de la chaîne de connexion pendant que le déclencheur s'exécute. Une seule instance de Automation peut donc utiliser la même configuration, mais utiliser différents serveurs de base de données pour les fonctionnalités relatives aux bases de données. Cela permet à l'utilisateur de configurer les déclencheurs dans les environnements de développement et de les exécuter dans des environnements de production sans rien changer à la configuration.

La logique de remplacement de la chaîne de connexion est définie dans le fichier **`DatabaseConnections.Config`** dans le dossier système de Automation.

1. Ouvrir le dossier système Automation `%PROGRAMDATA%\NiceLabel\NiceLabel 10`
2. Créer un fichier nommé **`DatabaseConnections.config`**.
3. Ouvrir le nouveau fichier créé DatabaseConnections.config dans un éditeur de texte et saisir le contenu de l'échantillon :

   ```
   <?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>
   ```

   Cet échantillon XMLL est un exemple, où un déclencheur existant contient une connexion au serveur Microsoft SQL appelé **`mySQLServer`** et à une base de données appelée **`myDatabase`**. La chaîne de connexion doit être mise à jour pour utiliser la base de données **`NEW_myDatabase`** sur le serveur **`NEW_mySQLServer`**.
   Il faut définir deux éléments, un pour changer le nom du serveur, et l'autre pour changer le nom de la base de données.

4. Vérifier la chaîne de connexion existante dans vos étiquettes, puis remplacer le contenu **`<Remplacement>`** par les valeurs correctes.
   Le fichier de configuration XML définit des paires source-destination. L'élément **`<Remplacement>`** contient un élément **`<De>`** et un élément **`<À>`** . Au cours de l'exécution du déclencheur, la chaîne De est remplacée par la chaîne À. Il est possible de définir autant d'éléments de **`<Remplacement>`** que nécessaire.

> **NOTE**
> Certaines chaînes de connexion utilisent Serveur=servername et d'autres Source de données=servername. Si vous utilisez la même base de données, il est inutile de saisir les noms de base de données.

5. Modifiez votre fichier **`product.config`** situé dans `%PROGRAMDATA%\NiceLabel\NiceLabel 10`. Ajouter l'élément `/Common/General/UseLocalReplacementStrings` et attribuer la valeur **`True`** :

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <Common>
        <General>
        ...
            <UseLocalReplacementStrings>True</UseLocalReplacementStrings>
        </General>
    </Common>
...
</configuration>
```

6. Redémarrer les deux services Automation après avoir ajouté ou modifié les fichiers de configuration.
7. Répéter les étapes 1 à 6 sur tous les ordinateurs qui exécutent les configurations Automation où vous souhaitez remplacer des chaînes de connexion de base de données.

> **NOTE**
> Le remplacement de chaînes de connexion de base de données ne modifie pas les chaînes de connexion dans vos fichiers d'étiquettes. Cette procédure ignore simplement les chaînes de connexion lors d'une impression depuis Automation.

9.12. Introduire des caractères spéciaux (Codes de Contrôle)

Les caractères spéciaux ou codes de contrôle sont des caractères binaires qui ne sont pas représentés sur le clavier. Impossible de les taper comme des caractères normaux, car ils doivent être codés par une syntaxe spéciale. Ces caractères sont utilisés pour communiquer avec des appareils en port série, recevant des données sur un port TCP/IP, ou pour travailler avec des fichiers binaires, tels que les fichiers d'impression.

Il y a deux méthodes pour introduire les caractères spéciaux :

- **Saisir manuellement les caractères** en utilisant l'un des exemples de syntaxes décrits :

- Utiliser la syntaxe **<caractère_spécial_acronyme>**, telle que **<FF>** pour FormFeed, ou **<CR>** pour Retour Chariot, ou **<CR><LF>** pour une nouvelle ligne.
- Utiliser la syntaxe **<#hex_code>**, telle que **<# 0D>** (décimale 13) pour Retour Chariot ou **<#00>** pour un caractère nul.

Pour plus d'informations, consulter l'article Liste des codes de contrôle.

- **Insérer les caractères de la liste.** Les objets pouvant comporter des caractères spéciaux affichent un bouton à flèche sur leur côté droit. Le bouton contient un raccourci vers tous les caractères spéciaux disponibles. Sélectionner un caractère dans la liste pour l'ajouter au contenu de l'objet. Pour plus d'informations, consulter l'article Utiliser des valeurs composées.

9.13. Liste des codes de contrôle

Code ASCII	Abréviation	Description
1	SOH	Début de l'entête
2	STX	Début de Texte
3	ETX	Fin de Texte
4	EOT	Fin de Transmission
5	ENQ	Requête
6	ACK	Accusé de réception
7	BEL	Sonnette
8	BS	Retour arrière
9	HT	Tabulation horizontale
10	LF	Nouvelle ligne
11	VT	Tabulation verticale
12	FF	Saut de page
13	CR	Retour chariot
14	SO	Basculement arrière
15	SI	Basculement avant
16	DLE	Fin des codes de contrôle
17	DC1	XON - Contrôle d'appareil 1
18	DC2	Contrôle d'appareil 2
19	DC3	XOFF - Contrôle d'appareil 3
20	DC4	Contrôle d'appareil 4
21	NAK	Accusé de réception Négatif
22	SYN	Veille Synchrone
23	ETB	Fin du bloc de Transmission

24	CAN	Annuler
25	EM	Fin de Média
26	SUB	Substituer
27	ESC	Échapper
28	FS	Séparateur de fichier
29	GS	Séparateur de Groupe
30	RS	Séparateur d'enregistrement
31	US	Séparateur d'unité
188	FNC1	Code de Fonction 1
189	FNC2	Code de Fonction 2
190	FNC3	Code de Fonction 3
191	FNC4	Code de Fonction 4

9.14. Attribution des licences et imprimantes utilisées

En fonction de la licence, le produit NiceLabel peut être limité au niveau du nombre d'imprimantes utilisables simultanément. Dans le cas d'une licence multi-utilisateurs, NiceLabel conserve la trace des numéros et des noms des différentes imprimantes utilisées pour imprimer sur tous les clients NiceLabel de votre environnement. L'identifiant unique de l'imprimante est une combinaison du nom du pilote (pas le nom de l'imprimante), de l'emplacement de l'imprimante et du port.

«Utiliser une imprimante» signifie que l'une des actions suivantes se trouve dans la configuration Automation:

- Imprimer l'étiquette
- Envoyer les données à l'imprimante
- Aperçu de l'étiquette
- Définir les paramètres d'impression

Chacune de ces actions signale qu'une imprimante a été utilisée. L'imprimante associée est ajoutée à la liste des imprimantes utilisées et reste dans cette liste pendant 7 jours. Pour supprimer une imprimante de la liste, ne pas l'utiliser pendant 7 jours. Elle sera supprimée automatiquement. Le logiciel affiche les informations concernant le **dernier travail** de telle sorte que l'échéance des 7 jours est visible pour chaque imprimante. Il est possible de bloquer une imprimante spécifique en cochant la case **Réservée**. La réservation garantit la disponibilité de l'imprimante à tout moment, même si une imprimante n'a pas fonctionné pendant 7 jours.

AVERTISSEMENT

Si le nombre d'imprimantes défini par la licence est dépassé, le logiciel entre dans une période de grâce de 30 jours. Pendant cette période, le nombre d'imprimantes autorisées est temporairement augmenté au double du nombre d'imprimantes attachées à la licence.

Cette période de grâce permet de résoudre le problème de licence sans panne d'impression ni perte de la possibilité de créer des étiquettes. Le nombre d'imprimantes autorisées dépassé est généralement une conséquence du remplacement d'imprimante dans votre environnement. Cela se produit si les anciennes et les nouvelles imprimantes sont utilisées simultanément ou si vous ajoutez de nouvelles imprimantes. Au terme des 30 jours, si le problème de licence n'est pas résolu, la quantité d'imprimantes disponibles revient au nombre attaché à la licence, en commençant dans la liste par les dernières imprimantes utilisées.

ASTUCE

Pour en savoir plus sur le mode de licence de NiceLabel 10, lire ce document – License NiceLabel 10.

9.15. Fonctionnement en mode service

NiceLabel Automation fonctionne comme un service Windows. Il est conçu pour ne pas exiger d'intervention durant le traitement des données et l'exécution des actions. Le service est configuré pour démarrer dès le lancement du système d'exploitation. Il fonctionne en arrière-plan tant que Windows fonctionne. NiceLabel Automation mémorise la liste de toutes les configurations chargées et des déclencheurs actifs. Le dernier état connu est rétabli automatiquement quand le serveur redémarre.

Le service fonctionne avec les droits du compte utilisateur sélectionné durant l'installation. Le service hérite de toutes les permissions d'accès de ce compte utilisateur, y compris pour les ressources réseau partagées, telles que les disques réseau et pilotes d'imprimantes. Utiliser le compte d'un utilisateur existant ayant des droits suffisants, ou mieux, créer un compte dédié pour NiceLabel Automation.

La gestion du service se fait en lançant les Services depuis le panneau de configuration Windows. Dans les systèmes d'exploitation Windows modernes, la gestion du service se fait aussi dans l'onglet Services du gestionnaire des tâches de Windows. Les Services permettent d'exécuter les tâches suivantes :

- Démarrer et arrêter les Services.
- Changer le compte sous lequel le service se connecte.

Les bonnes pratiques pour configurer le compte de service

- Même si c'est possible, nous déconseillons vivement d'exécuter le service sous le compte local système. C'est un compte local prédéfini par Windows avec des privilèges étendus sur l'ordinateur local, mais qui ne dispose généralement pas de privilèges d'accès aux ressources du réseau. Alors que NiceLabel Automation requiert l'accès complet au dossier `%temp%`, pas toujours disponible pour le compte du système local.
- Lors de la création d'un **nouveau compte utilisateur** pour le service NiceLabel Automation, il faut se connecter au moins une fois à l'ordinateur sur lequel le service tourne en utilisant ce nouveau compte. Cela garantit que le compte utilisateur est complètement créé. Par exemple, le dossier temporaire `%temp%` est créé au moment où le dossier est créé.
- Désactiver l'obligation de changer le mot de passe de temps en temps pour ce compte utilisateur.
- Il faut que le compte ait les permissions de **Se connecter en tant que service**.
- Exécuter le Service en mode 64 bits.

Accéder aux Ressources

NiceLabel Automation hérite de tous les droits du compte utilisateur Windows sous lequel le service s'exécute. Le service exécute toutes les actions sous le nom de ce compte. L'étiquette peut être ouverte si le compte a les permissions d'accès au fichier. L'étiquette peut être imprimée si le compte à accès au pilote d'imprimante.

Lors de l'utilisation du contrôle de version et des étapes d'approbation du Stockage de documents de Control Center, il faut que le service utilise un profil 'Impression-seule', tel qu'**Opérateur**. Ensuite, configurer les permissions d'accès pour le dossier spécifique en mode **lecture-seule** ou profil Opérateur. Ainsi, NiceLabel Automation n'utilise que les étiquettes approuvées, pas les brouillons.

Pour plus d'informations, consulter l'article Accès aux Ressources de Réseau Partagées.

Mode Service : 32 bits contre 64 bits

NiceLabel Automation peut fonctionner sur les systèmes natifs 32 bits (x86) et 64 bits (x64). Le mode d'exécution est déterminé automatiquement par le système d'exploitation Windows. NiceLabel Automation fonctionnera en mode 64 bits sur Windows 64 bits et fonctionnera en mode 32 bits sur Windows 32 bits.

- **Impression :** Parmi les avantages du fonctionnement de Automation en 64 bits, il y a la communication directe avec le service du spouleur d'imprimantes 64 bit sur Windows 64 bits. Ceci élimine les fameux problèmes avec SPLWOW64.EXE, qui est un 'intergiciel' permettant aux applications 32 bits d'utiliser le service de spouleur d'imprimantes 64 bits.
- **Accès à la base de données :** Pour fonctionner, le Service NiceLabel Automation 64 bits a besoin d'une version 64 bits des pilotes de la base de données pour accéder aux données. Pour plus d'informations, voir la section Accéder aux bases de données.

NOTE

Sans pilote de base de données 64 bits pour la base de données, il est impossible d'utiliser NiceLabel Automation en 64 bits. Il faut alors l'installer sur un système 32 bits, ou le forcer en mode 32 bits.

Forcer le mode d'Opération x86 sur Windows x64

Il peut y avoir des raisons d'utiliser NiceLabel Automation en application 32 bits sur Windows 64 bits.

Pour forcer NiceLabel Automation en mode x86 sur Windows x64, effectuer les opérations suivantes :

- Sélectionner Démarrer > Exécuter.
- Taper **regedit** et appuyer sur Entrée.
- Rechercher la clé

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\NiceLabelAutomationService2019
```

- Changer le nom de fichier en NiceLabelAutomationService10.x86.exe, en conservant le chemin existant.
- Redémarrer le service NiceLabel Automation.

AVERTISSEMENT

Il n'est pas recommandé de changer le mode service NiceLabel Automation. Si vous décidez quand même de le changer, assurez-vous d'effectuer un test de déclenchement complet avant de déployer la configuration dans l'environnement de production.

9.16. Ordre de recherche des fichiers

Quand NiceLabel Automation essaye de charger un fichier d'étiquette ou d'image, il tente de le localiser à différents endroits.

NiceLabel Automation Localise le fichier dans l'ordre suivant :

1. Vérifier si le fichier existe à l'endroit spécifié par l'action.
2. Vérifier si le fichier existe dans le même dossier que le fichier de configuration (.MISX).
3. Vérifier si le fichier de l'étiquette existe dans le dossier .\Labels (pour un graphique, il vérifie le dossier .\Graphics).
4. Vérifier si le fichier de l'étiquette existe dans le dossier ..\Labels (pour un graphique, il vérifie le dossier .\Graphics).
5. Vérifie si le fichier existe dans le dossier global d'étiquettes (dossier Graphics pour les graphiques) comme configuré dans les options.

Si le fichier n'existe dans aucun de ces endroits, l'action échoue et une erreur est signalée.

9.17. Sécuriser l'accès aux déclencheurs

Certains déclencheurs exigent un accès sécurisé aux déclencheurs. NiceLabel Automation permet d'activer des mesures de sécurité pour permettre à certains périphériques de confiance en réseau d'accéder aux déclencheurs. La configuration de la sécurité dépend du type de déclencheur. Certains types de déclencheurs permettent de configurer un accès sécurisé par défaut. Pour tous les déclencheurs basés sur le protocole TCP/IP, vous pouvez définir tous les détails dans le pare-feu Windows.

Configuration du pare-feu

Lors de l'utilisation de déclencheurs basés sur TCP/IP, tels que Déclencheur serveur TCP/IP, Déclencheur serveur HTTP ou Déclencheur Web Service s'assurer que les applications externes vous permettent de vous connecter aux déclencheurs. Chaque déclencheur fonctionne dans le service NiceLabel Automation, pour lequel l'accès est gouverné par le pare-feu Windows.

> **NOTE**
> Par défaut, le pare-feu Windows est configuré pour permettre toutes les connexions entrantes au service NiceLabel Automation. Cela rend la configuration et l'essai des déclencheurs plus aisée, mais laisse la porte ouverte.

Si le déploiement de NiceLabel Automation dans l'entreprise est soumis à des règles de sécurité strictes, il faut adapter les règles du pare-feu en conséquence.

Par exemple :

- Affiner le pare-feu pour accepter seulement le trafic entrant de sources bien connues.
- Autoriser les données entrantes seulement sur les ports prédéfinis.
- Permettre la connexion de certains utilisateurs seulement.
- Définir sur quelles interfaces les connexions entrantes sont acceptées.

Pour effectuer des changements au pare-feu Windows, ouvrez la console de gestion **Pare-feu Windows avec Sécurité Avancée** dans le **Panneau de Contrôle** > **Système et Sécurité** > **Pare-feu Windows** > **Paramètres Avancés**.

> **NOTE**
> Quand NiceLabel Automation est lié aux produits NiceLabel Control Center, il faut activer la connexion entrante sur le port **56415/TCP**. Si ce port est fermé, NiceLabel Automation ne peut pas être géré par le Control Center.

Permettre l'Accès Basé sur les Permissions d'Accès au Fichier

Le déclencheur de fichier va s'exécuter sur changement de l'horodatage dans le ou les fichiers surveillés. Il faut mettre les fichiers du déclencheur dans un dossier accessible au service NiceLabel Automation. Le compte utilisateur sur lequel le Service fonctionne doit pouvoir accéder aux fichiers. Simultanément, les permissions d'accès aux emplacements déterminent quel utilisateur et ou application peut sauvegarder le fichier du déclencheur. Il faut paramétrer les permissions d'accès de façon à ce que seuls les utilisateurs autorisés puissent enregistrer les fichiers.

Permission d'accès basée sur l'adresse IP & le nom d'hôte

Vous pouvez protéger l'accès au déclencheur du serveur TCP/IP avec deux listes d'adresses IP et noms d'hôtes.

- La première liste **'Autoriser les connexions des hôtes suivants'** contient les adresses IP ou noms des périphériques qui peuvent envoyer des données au déclencheur. Si un appareil a une adresse IP de cette liste, il est autorisé à envoyer des données au déclencheur.
- La seconde liste **'Refuser les connexions pour les hôtes suivants'** contient des adresses IP ou noms de périphériques qui ne sont pas autorisés à envoyer des données. Si un appareil a une adresse IP de cette liste, il n'est pas autorisé à envoyer des données au déclencheur.

Permettre l'accès sur base des noms d'utilisateurs & mots de passe

Protéger l'accès au déclencheur de Serveur HTTP en activant l'authentification d'utilisateur. Quand activée, chaque requête HTTP envoyée au déclencheur serveur HTTP doit comprendre la combinaison **'nom et mot de passe utilisateur'** qui correspond à la combinaison définie.

Permettre l'accès en fonction de l'appartenance à un Groupe d'application

Protéger l'accès au déclencheur de Serveur HTTP en ajoutant des utilisateurs au groupe d'application dans Control Center. Quand cette option est activée, seuls les membres authentifiés sont autorisés à accéder au déclencheur.

9.18. Session d'impression

Une session d'impression permet d'imprimer plusieurs étiquettes dans un seul travail d'impression. Quand une session d'impression est activée, l'imprimante reçoit, traite et imprime d'un coup toutes les étiquettes du travail d'impression. Ainsi, la vitesse d'impression augmente en raison du traitement continu du paquet d'étiquettes.

ASTUCE

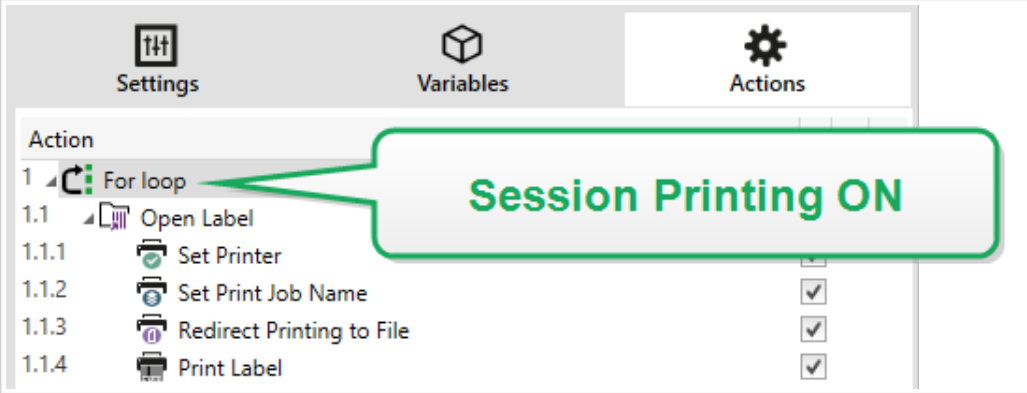
La session d'impression est l'alternative à l'impression sans session qui envoie chaque étiquette à l'imprimante par des travaux d'impression séparés.

NOTE

Automation active automatiquement la session d'impression en fonction de la configuration des actions.

Comment démarre l'impression en session ?

La session d'impression démarre automatiquement quand les actions Boucler , Pour chaque enregistrement ou Pour chaque ligne sont présentes dans le flux de travail. Dans ce cas, l'action indentée Imprimer l'étiquette active automatiquement une session d'impression. Toutes les actions Imprimer concernant tous les articles à boucler sont inclus dans un unique travail d'impression.



Comment finit la session d'impression ?

Chaque session d'impression se termine soit avec la dernière boucle, soit avec l'action Imprimer l'étiquette associée à l'une des conditions suivantes :

- L'imprimante change. Quand une autre imprimante est sélectionnée par l'action Définir l'imprimante, la session d'impression se termine.
- Le port de l'imprimante change. Quand le travail d'impression est redirigé vers un fichier par l'action Rediriger l'impression vers un fichier, la session d'impression se termine.
- L'étiquette change. Quand une autre étiquette à imprimer est sélectionnée par l'action Ouvrir l'étiquette, la session d'impression se termine.
- Une commande personnalisée est envoyée pour terminer la session d'impression. Quand une commande `SESSIONEND` est envoyée par l'action Envoyer une commande personnalisée, la session d'impression se termine.

> **NOTE**
> Dans ce cas, `SESSIONEND` doit être le seul élément envoyé dans l'action Envoyer une commande personnalisée. Pour envoyer d'autres commandes, utiliser différentes actions Envoyer une commande personnalisée.

> **NOTE**
> Des configurations plus complexes peuvent comporter plusieurs boucles indentées les unes sous les autres. Dans ce cas, la session d'impression se termine quand la boucle parent la plus éloignée sort.

9.19. Conseils et astuces d'utilisation de variables dans les actions

Si vous utilisez des variables dans les actions NiceLabel Automation, suivez ces recommandations.

- **Placer les variables entre parenthèses.** Si des variables contenant des espaces dans leur nom se réfèrent à des variables dans des actions, telles que Exécuter une requête SQL ou Exécuter un Script, il faut mettre les variables entre parenthèses, ex. : **`(Nom Produit)`**. Utiliser également des parenthèses si le nom de la variable est un nom réservé, par exemple dans les requêtes SQL.
- **Placer deux points devant le nom de variable.** En ce qui concerne les variables de l'action Exécuter une requête SQL ou dans Déclencheur de base de données, il faut placer deux points (`:`) devant le nom de la variable, par exemple **`:(Product ID)`**. L'analyseur SQL le comprendra comme 'valeur variable'.

```
SELECT * FROM MyTable WHERE ID = :(ProductID)
```

- **Convertir des valeurs en nombre entier pour le calcul.** Si vous voulez effectuer un calcul numérique avec les variables, il faut convertir la variable en nombre entier. Définir les variables en numérique limite seulement les caractères acceptés pour la valeur mais ne change pas le type de la variable. NiceLabel Automation traite toutes les variables de type chaîne de caractères. Dans VBScript, utiliser la fonction **`CInt()`**.
- **Paramètres par défaut/valeurs de démarrage des scripts.** Pour utiliser des variables dans une action, il faut qu'elles aient une valeur par défaut, autrement la vérification du script échouera. Définir les valeurs par défaut dans les propriétés des variables ou dans le script (et les enlever après avoir testé le script).

9.20. Mode de Traçage

Par défaut, NiceLabel Automation enregistre les événements dans la base de données du journal. Cela inclut la collecte d'informations de haut niveau, telles que :

- le journal d'exécution des actions
- le journal d'exécution des filtres
- le journal des mises à jour des états des déclencheurs

Pour plus d'informations, consulter l'article Options de journalisation des événements.

Toutefois, le journal par défaut n'enregistre pas les détails minimes des exécutions. Le mode de traçage doit être activé quand une résolution de problème en profondeur est requise pour l'exécution du code. Dans ce mode, NiceLabel Automation enregistre les détails de toutes les exécutions internes qui se déroulent durant le traitement du déclencheur. Le mode de traçage ne doit être activé qu'aux fins de résolution de problèmes pour collecter les données. Il faut ensuite le désactiver pour reprendre le mode d'opération normal.

AVERTISSEMENT

Le mode de traçage ralentit le traitement. Il ne peut être utilisé que si c'est une instruction de l'équipe d'assistance technique de NiceLabel.

Activer le mode de traçage

Pour activer le mode de traçage, effectuer les opérations suivantes :

1. Rechercher le dossier système de NiceLabel Automation.

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10
```

2. Créer une copie de sauvegarde du fichier **`product.config`**.
3. Ouvrir le fichier **`product.config`** dans un éditeur de texte. Le fichier a une structure XML.
4. Ajouter l'élément **`Common/Diagnostics/Tracing/Enabled`** et lui assigner la valeur **Vrai**.
   Le fichier doit avoir le contenu suivant :

```xml
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <Common>
        <Diagnostics>
            <Tracing>
                <Enabled>True</Enabled>
                <Folder>c:\Troubleshooting\TracingLogs</Folder>
            </Tracing>
        </Diagnostics>
    </Common>
...
</configuration>
```

5. Après la sauvegarde du fichier, le Service NiceLabel Automation applique les paramètres automatiquement.
6. Par défaut, les fichiers de traçage (extension en *.LOG) apparaissent dans le même dossier système.
   Pour remplacer le dossier du journal, utiliser le fichier **`product.config`**. Spécifier le dossier du journal personnalisé dans l'élément **`Dossier`**. Cet élément est facultatif.
7. Démarrer Automation Manager pour confirmer que le mode de traçage est activé. Activé, le mode de traçage affichera le texte **Le traçage a été activé** dans l'interface de notification au-dessus de la liste de déclencheurs.

Après avoir reproduit le problème avec le traçage activé, désactiver le traçage dans le fichier `product.config` ou restaurer votre fichier `product.config` original.

```
            <Tracing>
                <Enabled>False</Enabled>
```

9.21. Comprendre les paramètres d'imprimante et DEVMODE

> **NOTE**
> La structure de données DEVMODE fait partie de la Structure GDI d'impression API de Windows. Cet article comporte un contenu très technique, pertinent uniquement pour des obligations spécifiques.

À chaque impression d'étiquette avec le logiciel NiceLabel (ou de document dans une application Windows appropriée), l'application d'impression va lire les paramètres de l'imprimante définis dans le pilote d'imprimante et les appliquer au travail d'impression. La même étiquette peut être imprimée sur différentes imprimantes en sélectionnant un pilote d'imprimante différent. Les paramètres de la nouvelle imprimante s'appliquent chaque fois à la nouvelle étiquette sélectionnée.

L'impression d'un document de texte sur l'une ou l'autre imprimante laser produit généralement un résultat identique ou comparable. L'impression d'étiquettes utilisant deux imprimantes d'étiquettes différentes peut produire des résultats très incohérents. Pour produire des résultats comparables, le même fichier d'étiquette doit demander des paramètres de pilote d'imprimante additionnels, tels que l'ajustement des décalages, la vitesse et le chauffage d'impression. NiceLabel applique aussi les paramètres d'imprimante à chaque impression. Par défaut, les paramètres d'imprimante sont enregistrés dans le fichier d'étiquette pour l'imprimante sélectionnée.

Qu'est-ce que le DEVMODE ?

Le DEVMODE est une structure Windows qui contient les paramètres d'imprimante (informations d'initialisation et d'environnement de l'imprimante). Il est constitué de deux parties : public et privé. La partie publique contient les données communes à toutes les imprimantes. La partie privée contient les données spécifiques à une imprimante donnée. La partie privée peut avoir une longueur variable et contenir tous les paramètres spécifiques d'un fabriquant.

- **Partie publique :** Cette partie encode les paramètres généraux exposés dans le modèle du pilote d'imprimante, tels que le nom d'imprimante, la version du pilote, la taille du papier, l'orientation, la couleur, duplex et similaires. La partie publique est la même pour tout pilote d'imprimante et ne supporte pas les spécificités des imprimantes d'étiquettes (imprimantes thermiques, imprimante à jet d'encre industrielles, machines de gravure laser).
- **Partie privée :** Cette partie encode les paramètres qui ne sont pas disponibles dans la partie publique. Les pilotes d'imprimante NiceLabel utilisent cette partie pour sauvegarder les données spécifiques au modèle, telles que la vitesse d'impression, les paramètres de chauffage, les décalages, le mode d'impression, le type de média, capteurs, couteaux, encodage graphique, support RFID etc. La structure des données dans la partie privée du DEVMODE est un flux de données binaires définies par les développeurs du pilote.

Modifier le DEVMODE

La structure des données du DEVMODE est stockée dans les registres Windows. Il y a deux copies de la structure : les paramètres d'imprimante par défaut et les paramètres d'imprimante spécifiques à l'utilisateur. Le DEVMODE (paramètres d'imprimante) peut être modifié en changeant les paramètres dans le pilote d'imprimante. Les deux premières options sont liées à Windows, la troisième option est disponible dans le logiciel NiceLabel.

- **Paramètres de l'imprimante par défaut :** Ces paramètres sont définis dans **Propriétés de l'imprimante** > **Onglet Avancé** > **Impression par défaut**.
- **Paramètres utilisateur spécifiques :** Ils sont sauvegardés séparément pour chaque utilisateur dans les clés de registres utilisateurs HKEY_CURRENT_USER. Par défaut, les paramètres utilisateur spécifiques sont hérités des paramètres d'imprimante par défaut. Les paramètres utilisateur spécifiques sont définis dans **Propriétés de l'imprimante > Préférences**. Toutes les modifications apportées ici n'affectent que l'utilisateur actuel.
- **Paramètres spécifiques d'étiquette :** L'auteur de l'étiquette qui utilise le logiciel NiceLabel peut incorporer le DEVMODE dans l'étiquette. Ceci permet de déplacer les paramètres d'imprimante. Quand l'étiquette est copiée sur un autre ordinateur, les paramètres d'imprimante la suivent. Pour incorporer les paramètres dans l'étiquette, activer l'option **Utiliser les paramètres personnalisés sauvegardés dans l'étiquette** dans *Fichier > Paramètres Étiquette > Onglet Imprimante* dans Designer Pro. Les paramètres d'imprimante sont modifiables dans l'étiquette en sélectionnant le panneau *Imprimante* dans les *Propriétés du document*.

Appliquer le DEVMODE personnalisé à l'impression

Dans NiceLabel Automation, ouvrir un fichier d'étiquette et lui appliquer le DEVMODE personnalisé. À l'impression de l'étiquette, le masque de l'étiquette est pris dans le fichier .NLBL et le DEVMODE applique le formatage spécifique lié à l'imprimante. Ainsi, il n'y a qu'une seule étiquette maître. Dans ce cas, l'impression sera la même, quelle que soit l'imprimante utilisée, car les paramètres optimaux d'impression sont appliqués pour cette imprimante.

Pour appliquer un DEVMODE personnalisé à l'étiquette, il y a deux options :

1. Utiliser une action, plus spécifiquement les **Paramètres d'imprimante**.
2. Le fichier de commande JOB, plus spécifiquement la commande **SETPRINTPARAM** avec le paramètre **PRINTERSETTINGS**. Pour plus d'informations, consulter l'article Utiliser des commandes personnalisées.

9.22. Utiliser le même compte utilisateur pour configurer et exécuter les déclencheurs

Le Service NiceLabel Automation s'exécute toujours sous les infos d'identification du compte utilisateur configuré pour le service. Toutefois, Automation Builder s'exécute toujours sous les infos de l'utilisateur connecté. Les infos d'utilisateur du compte de service et du compte connecté peuvent être différentes.

Même si l'aperçu du déclencheur s'affiche sans erreurs dans Automation Builder, le Service peut rapporter un message d'erreur en raison des différences d'informations concernant les utilisateurs. Si l'utilisateur connecté a la permission d'accéder aux dossiers et imprimantes, le compte utilisateur du service utilisé peut ne pas l'avoir.

Tester l'exécution des déclencheurs dans Automation Builder en utilisant les mêmes infos d'utilisateur que celles du service. Pour cela, utiliser Automation Builder sous le même compte utilisateur que celui défini pour le service.

Pour utiliser Automation Builder sous un autre compte, procéder comme suit :

1. Appuyer et maintenir la touche **Maj**, ensuite **cliquer à droite** sur l'icône Automation Builder.
2. Sélectionner **Utiliser sous un autre compte utilisateur**.
3. Saisir les infos d'utilisateur identiques à celles définies dans le service NiceLabel Automation.
4. Cliquer sur **OK**.

Pour utiliser fréquemment Automation Builder avec les infos de l'autre compte utilisateur, voir l'utilitaire en ligne de commande Windows **RUNAS**. Utiliser `/user` pour spécifier le compte utilisateur, et `/savecred`. Ce dernier vous permet de saisir le mot de passe une seule fois, puis le mémorise pour la prochaine fois.