NiceLabel Automation 2017
Guide d'Utilisation

www.nicelabel.com

1 Contenus

2 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. Les applications utilisées dans toutes les divisions et sur tous les sites d'une entreprise peuvent désormais imprimer des masques d'étiquettes approuvés.

NiceLabel Automation représente le système idéal pour imprimer des étiquettes en synchronisant les événements de l'activité industrielle avec la production d'étiquettes. L'impression automatique sans interaction humaine est de loin la façon la plus efficace pour éliminer les erreurs humaines et maximaliser les performances.

L'impression automatique d'étiquettes utilisant une application basée sur des déclencheurs comporte 3 processus principaux.

Déclencheur

Les déclencheurs sont une fonction simple mais puissante qui permettent d'automatiser le travail. En fait, 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 sont utiles pour les travaux 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. Quand 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 Fichier
- Déclencheur port série
- Déclencheur Base de Données
- Déclencheur TCP/IP
- Déclencheur HTTP
- Déclencheur Web Service

Extraction de données et Placement

Dès que l'impression est activée, NiceLabel Automation extrait les données d'étiquette et les insère dans les champs variables de l'étiquette.

Les Filtres d'extraction de données peuvent utiliser :

- Des fichiers de Texte Structurés
- Des fichiers de Texte Non-Structurés
- Différents fichiers XML
- Des données binaires : remplacement d'imprimante, export d'un ancien logiciel, données de périphériques, etc.

Exécution d'Action.

Quand les données correspondent aux champs variables de l'étiquette, NiceLabel Automation effectue les actions. Les opérations de base comprennent généralement les actions **Ouvrir Étiquette** et **Imprimer Étiquette** pour imprimer les données extraites sur les étiquettes. Les données peuvent aussi être envoyées à des emplacements spécifiques : des fichiers sur le disque, à un serveur Web, à des périphériques, etc. En tout, il y a le choix entre 30 actions différentes.

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

3 Conventions Typographiques

Le texte qui apparaît en **gras** se réfère aux noms et boutons du menu.

Le texte qui apparaît en *italique* se réfère aux options, actions de confirmation telle que Lecture seule et aux emplacements tels que Dossier.

Le texte encadré par les signes <Plus-petit et Plus-grand> Se réfère aux touches du clavier de l'ordinateur tel que `<Enter>`.

Les variables sont entourées de [crochets].

NOTE: Ceci représente une note.

EXEMPLE: Ceci représente un exemple.

Ceci représente une bonne pratique.

ATTENTION : Ceci représente un avertissement.

CONSEIL: Ceci représente un conseil.

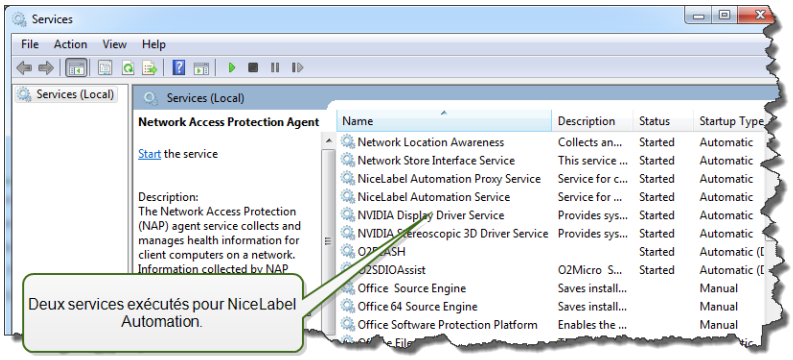
4 Paramétrage de l'application

4.1 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 est composé de trois éléments.

- **Automation Builder.** C'est une application de configuration que le développeur utilisera pour créer les déclencheurs, filtres et actions à exécuter quand les données sont reçues par le déclencheur. Cette application s'exécute toujours comme une application 32-bits.
- **Automation Manager.** Ceci est l'application de gestion utilisée pour la surveillance de l'exécution des déclencheurs en temps réel et démarrer/arrêter les déclencheurs. Cette application s'exécute toujours comme une application 32 bits.
- **NiceLabel Automation Service.** C'est le moteur d'impression qui exécute les règles définies dans les déclencheurs. Il y a en fait deux applications de service, le service NiceLabel Automation et le service ProxyNiceLabel. 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.



4.2 Spécifications Système Requises

- CPU : Processeur Intel ou compatible de la famille x86
- Mémoire : 2 GB ou plus de RAM
- Disque dur : 1GB de place disponible
- Un des Systèmes d'exploitation Windows 32 ou 64 bits: Windows Server 2008 R2, Windows 7, Windows 8, Windows 8.1, Windows Server 2012, Windows Server 2012 R2, Windows 10m Windows Server 2016(Windows Server Core et Windows Nano Server

ne sont pas utilisables)

- Microsoft .NET Framework Version 4.5
- Écran : Résolution de l'écran 1366×768 ou plus élevée
- Editeur d'étiquette :
  - Recommandé: NiceLabel V2017 (fichier au format .NLBL)
  - Minimum: NiceLabel Pro V5.4 (fichier au format .LBL) mais avec possibilité d'incompatibilités
- Pilotes d'imprimantes recommandés : NiceLabel Printer Drivers V5.1 ou supérieure
- Accès complet au 'Dossier Système' de l'application, dans lequel les évènements sont sauvegardés dans une base de données.

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

- Accès complet aux dossier du compte utilisateur du service `%temp%`.

4.3 Installation

NOTE: Ci-dessous une description sommaire de la procédure d'installation. Pour plus d'informations, consulter le **Guide d'Installation**.

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

Pour installer NiceLabel Automation, effectuer les opérations suivantes :

1. Insérer le DVD NiceLabel

   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 l'**installation** de **NiceLabel**.
3. Suivre les instructions de l'**Assistant d'installation**.

   Durant l'installation, l'interface vous proposera de saisir le nom de l'utilisateur sous lequel le service NiceLabel Automation fonctionnera. Il faut donner le nom d'un utilisateur réel, car le service héritera des privilèges de cet utilisateur. Pour plus d'informations, consulter l'article Fonctionnement en mode service.

Mise à jour

Pour effectuer la mise à jour d'une mouture de NiceLabel Automation dans la même version, 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, en gardant les paramètres

existants. La base de données du journal sera vidée durant la mise à jour.

NOTE: Deux versions majeures d'un même produit NiceLabel s'installent côte à côte.

4.4 Activation

Il faut activer le logiciel NiceLabel Automation pour permettre l'exécution des déclencheurs configurés. La procédure d'activation requiert une connexion Internet, de préférence sur la machine où le logiciel est installé. La même procédure d'activation est utilisée pour activer la clé de licence d'essai.

NOTE: Le logiciel peut être activé indifféremment avec Automation Builder ou avec Automation Manager.

Activation dans Automation Builder

1. Exécuter **Automation Builder**.
2. Sélectionner **Fichier>A propos>Activer la licence**.
   L'assistant d'activation va démarrer.
3. Suivre les instructions à l'écran.

Activation dans Automation Manager

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

4.5 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 dispose des mêmes fonctionnalités que la version avec licence, ce qui permet d'évaluer le produit avant de l'acheter. Automation Manager affichera la notification d'essai en permanence ainsi que le nombre de jours restant pour l'essai. Quand le mode d'essai expire, le service NiceLabel Automation n'exécute plus 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, consulter l'article Activation.

4.6 Onglet Fichier

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

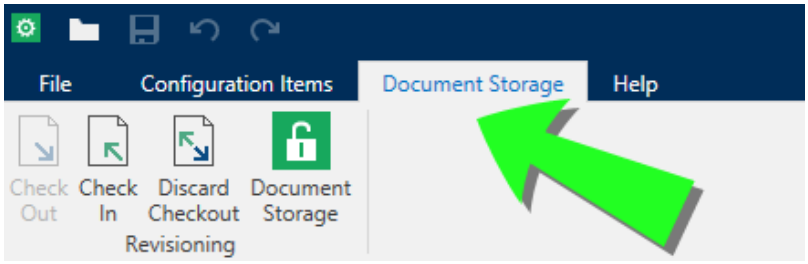
- **Nouveau:** pour créer un nouveau fichier de configuration.
- Ouvrir: pour ouvrir les fichiers de configuration existants.
- **Ouvrir un fichier NiceWatch :** ouvre une ancienne configuration NiceWatch de NiceLabel.
- Enregistrer: enregistre le fichier de configuration actif.
- Enregistrer sous: permet 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.

4.6.1 Ouvrir

La boite 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 document du NiceLabel Control Center connecté. Si la gestion des versions est activée sur cette emplacement du Control Center, d'autres menus s'ouvrent. L'onglet **Stockage de documents** permet de Gérer la copie du fichier de configuration enregistré.



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

4.6.2 Compatibilité Avec Les Produits NiceWatch

NiceLabel Automation peut charger les configurations qui ont été définies dans un des 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'étiquette. 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, la notification concernera :

- **La compatibilité avec la configuration du déclencheur.** A 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. Certaine sont totalement indisponibles, d'autres sont configurées différemment. Si le fichier MIS contient des fonctionnalités incompatibles, elles figureront dans une liste et seront supprimées 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 de NiceLabel Automation pour recréer la configuration.

- **La compatibilité avec le masque de l'étiquette.** Si les masques d'étiquettes existants contiennent des fonctionnalités non supportées par le moteur d'impression de NiceLabel Automation, il y aura des messages d'erreur dans le panneau du 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 la Base de connaissances - article KB251.

Ouvrir la configuration NiceWatch pour l'éditer

Ouvrir la configuration NiceWatch existante (fichier .MIS) dans Automation Builder et l'éditer dans Automation Builder. La configuration peut seulement être enregistrée au format .MISX.

Pour éditer la configuration NiceWatch, effectuer les opérations suivantes :

1. Démarrer Automation Builder.
2. Sélectionner **Fichier>Ouvrir fichier NiceWatch**.
3. Dans la boite de dialogue Ouvrir, rechercher le fichier de configuration NiceWatch (fichier .MIS).
4. Cliquer sur **OK**.
5. Si la configuration contient des fonctionnalités non-supportées, elle en affichera la liste. Elles seront retirées 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, il 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 l'interface 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 le bouton **Démarrer**.

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

4.6.3 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 boite de dialogue **Enregistrer sous**.

4.6.4 Enregistrer Sous

Enregistrer sous: permet d'enregistrer le fichier de configuration actif en donnant son nom et son emplacement.

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

4.6.5 Options

Utiliser les paramètres de cette boite de dialogue pour personnaliser l'application. Sélectionner le groupe dans le panneau de gauche puis configurer les paramètres dans le panneau de droite.

Dossiers

Il est possible de choisir des dossiers par défaut pour stocker les étiquettes, formulaires, bases de données et fichiers graphiques. Par défaut, l'emplacement du dossier est dans les Documents de l'utilisateur. Ce sont les dossiers par défaut dans lesquels NiceLabel Automation recherche les fichiers pour lesquels il ne connaît que le nom et pas le chemin. Pour plus d'informations, consulter l'article Ordre de recherche des fichiers requis.

Les modifications de dossiers se propagent au Service en une minute. Pour appliquer les modifications immédiatement, redémarrer le service NiceLabel Automation.

NOTE: Les paramètres appliqués ici sont enregistrés dans le profil de l'utilisateur connecté. Si le Services NiceLabel Automation fonctionnent sous un autre compte utilisateur, il faut d'abord se connecter à Windows en utilisant cet autre compte, ensuite changer le dossier d'étiquette par défaut. On peut aussi utiliser l'utilitaire Windows en ligne de commande RUNAS pour lancer Automation Builder sous cet autre utilisateur.

Langue

L'onglet Langue permet de choisit la langue de l'interface de NiceLabel Automation. Sélectionner la langue désirée et cliquer sur **OK.**

NOTE: Le changement s'appliquera au redémarrage de l'application.

Variables Globales

L'onglet Variables globales permet de définir l'emplacement à utiliser pour stocker les variables globales:

- **Utiliser les variables globales stockées sur le serveur (Control Center)** Détermine un emplacement pour stocker la variable globale sur le Control Center.

  NOTE: Cette option est disponible avec une licence de NiceLabel LMS.

- **Utiliser les variables globales stockées dans un fichier (local ou partagé)** Détermine un emplacement pour stocker la variable globale dans un dossier local ou partagé. Saisir le chemin exact ou cliquer sur **Ouvrir** pour localiser le fichier.

Control Center

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

NOTE: Cet onglet n'est disponible qu'après activation de la licence LMS.

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

- **Imprimer les événements:** enregistre les événements d'impression du poste de travail.
- **Survenance d'erreurs:** enregistre toutes les erreurs rapportées.

NOTE: Par défaut toutes les impressions et les erreurs sont inscrites dans le journal de 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.
- Que le pilote d'imprimante utilisé soit un pilote NiceLabel.

Imprimantes utilisées

NOTE: Le journal des imprimante utilisées est disponible avec une licence multi sites.

L'onglet **Imprimantes utilisées** affiche le journal de l'utilisation des imprimantes installées. Il fournit les renseignements concernant les imprimantes utilisées pour l'impression.

Le groupe **Information d'utilisation de l'imprimante** affiche la quantité de ports d'imprimante autorisés utilisée pour imprimer sur plusieurs imprimantes.

- **Nombre d'imprimantes autorisées par licence.** Nombre d'imprimantes utilisables avec la licence Designer.
- **Nombre d'imprimantes utilisées au cours des 7 derniers jours:** Nombre d'imprimantes utilisées par le Designer au cours des 7 derniers jours.

CONSEIL: Pendant une période de 7 jours la licence permet d'utiliser uniquement le nombre spécifié d'imprimantes différentes.

ATTENTION : Lorsque le nombre d'imprimantes autorisées est dépassé - ce nombre est défini par la licence - une alerte apparaît. Lorsque le nombre d'imprimantes autorisées est doublé, l'impression n'est plus permise.

Les status de l'impression sont visibles en plusieurs colonnes:

- **Imprimante** Nom et modèle de l'imprimante sélectionnée pour le travail d'impression.

  NOTE: Quand l'imprimante est partagée, seul le modèle s'affiche.

- **Emplacement** Nom de l'ordinateur qui a envoyé le travail d'impression.
- **Port.** Port utilisé par l'imprimante.
- **Dernier travail** Temps écoulé depuis le dernier travail d'impression
- **Conservée** Empêche la suppression d'une imprimante inutilisée pendant plus de 7 jours.

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

Le groupe **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, seule les imprimantes réservées sont autorisées pour éditer et imprimer les étiquettes dans NiceLabel 2017.

  CONSEIL: 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'étiquetage thermiques ou laser dédiées et limiter l'impression à ces seules imprimantes pour garantir la continuité de l'impression d'étiquettes avec une licence multi utilisateurs.

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

  1. Parcourir le dossier système.

     EXEMPLE: [[[Undefined variable Variables.Path-System-Designer]]]

  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 les lignes suivantes:

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

  5. Enregistrer le fichier. Le `Example Printer` est réservé.

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 convient pas pour l'ordinateur, en sélectionner un autre. Attention, 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 démarrage du nettoyage du journal. A l'heure définie, la base de données du journal sera purgée des événements anciens.
- **Effacer les entrées du journal lorsqu'elles ont plus de (jours)** Spécifie la durée de détention des événements dans la base de données du journal. Tous les événements plus vieux que le nombre de jours spécifié seront purgés de la base de données lors du nettoyage de la base de données du journal.
- **Messages du journal.** Spécifie le type de messages à enregistrer dans le journal. Durant les phases de développement et de test, il vaut mieux activer une journalisation détaillée. Activer tous les messages permet un traçage plus aisé de l'exécution des déclencheurs. Mais, pendant la phase de production il vaut mieux réduire la quantité de messages pour activer seulement la journalisation des erreurs.

Performance

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

- **Mettre en cache les fichiers distants.** Pour améliorer le temps de sortie de la première étiquette et les performances générales, NiceLabel Automation permet la mise en cache de fichiers. Lors du chargement 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 puisse commencer.

  CONSEIL: Quand la mise en cache local est activée, l'effet de latence du réseau est réduit puisque les fichiers d'étiquettes et images sont déjà chargé .

  Automation utilise le dossier local suivant comme cache pour les fichiers distants `:%PROGRAMDATA%\NiceLabel\NiceLabel 2017\FileCache`.

  - **Actualiser les fichiers cache (minutes).** Définit l'intervalle de temps en minutes durant lequel le cache sera synchronisé avec les fichiers dans leur dossier d'origine. C'est l'intervalle de temps durant lequel le système est autorisé à utilise une version qui n'est pas forcément la dernière.
  - **Supprimer les fichiers du cache quand ils ont plus de (jours)** Définit tous les combien (en jours) les fichiers sont supprimés du cache.

NOTE: La mise en cache de fichiers est compatible avec les formats de fichiers image et étiquette. Après avoir activer la mise en cache des fichiers, redémarrer le service Automation pour que la modification prennent effet.

Utilisation en cluster

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

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

4.6.6 A Propos

La boite de dialogue A propos fournit les renseignements 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 les détails sur le logiciel et permet de changer de niveau de produit NiceLabel 2017.

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

- Les informations sur la **durée du mode test:** avec le nombre de jour restant 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 boite de dialogue d'activation de la licence NiceLabel 2017. Consulter le guide d'installation de NiceLabel 2017 pour plus de détail 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 2017 n'est plus activé.
- **Changer le niveau du produit:** Ouvre la boite 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 boite de dialogue de mise à jour du niveau de produit. Consulter le guide d'installation de NiceLabel 2017 pour plus de détail sur le processus de mise à jour de la licence.

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

5 Comprendre les filtres

5.1 Comprendre Les Filtres

NiceLabel Automation utilise des filtres pour définir la structure des données reçues par les déclencheurs. Chaque fois qu'un déclencheur reçoit une donnée, cette donnée est analysée par un ou plusieurs filtres qui extraient les données nécessaires. Chaque filtre est configuré avec des règles décrivant la façon d'identifier les champs dans les données.

> 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 articles Filtre de Texte Structuré, Filtre de données non-structurées et Filtre XML.

Structure de Données

La complexité du filtre dépend de la structure des données. Les données qui sont déjà sous une forme structurée, telles que CSV ou XML peuvent être extraites facilement . Dans ce cas, les noms des champs sont déjà définis dans les données. L'extraction des paires `nom-valeur` est rapide. Quand les données n'ont pas une structure claire, la définition des règles d'extraction prend plus de temps. Ces données peuvent se présenter sous forme de documents exportés ou de rapports provenant d'un vieux système, de communication interceptée entre deux périphériques, d'une capture d'un flux d'impression etc.

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

NiceLabel Automation supporte différents types de données d'entrée qui peuvent tous être analysées par l'une des méthodes de filtrage. Il faut choisir le filtre qui correspond au type de données entrantes. Par exemple, le **Filtre de Texte Structuré** pour des données CSV et le **Filtre XML** pour les données XML. Pour toutes données non-structurées, utiliser le **Filtre de Données Non-structurées**. Pour plus d'informations, consulter 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. L'action va exécuter les règles du filtre sur les valeurs et extraire les valeurs.

Chaque déclencheur peut exécuter autant d'actions "Utiliser un Filtre de Données" que nécessaires. Si ces données reçues sont complexes et ne peuvent pas être analysées par un seul filtre, il faut définir et exécuter les règles de plusieurs filtres dans des actions "Utiliser un Filtre de Données". Elles seront exécutées l'une après l'autre. A la fin, utiliser, pour une même étiquette, les valeurs extraites par toutes les actions.

Associer les Champs aux Variables

Pour utiliser les valeurs extraites, il faut les enregistrer dans des variables. L'action "Utiliser un Filtre de Données" n'extrait pas seulement les valeurs, mais elle les enregistre aussi dans des variables. Pour configurer ce processus, relier la variable au champ respectif. La valeur du champ sera ensuite enregistrée dans la variable associée.

Il est conseillé de définir les champs et variables avec les mêmes noms. Dans ce cas, l'outil automatique va associer les variables aux champs de mêmes noms, éliminant le processus manuel.

Le mappage automatique est disponible pour tous les types de filtres. Quand le mappage automatique est activé, l'action "Utiliser un Filtre de Données" va extraire automatiquement les valeurs et les associer aux variables de mêmes noms que les champs. Pour plus d'informations, consulter les articles Activer la Structure Dynamique pour les filtres de Texte Structuré, Définition des zones d'affectation pour les filtres de Données Non-structurées et Définition de la zone d'affectation XML pour les filtres XML.

Définition des Actions à Exécuter pour les Données Extraites

Généralement il faut exécuter des actions avec les données extraites, telles que **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 un 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 sera aussi exécutée 5 fois, une fois pour chaque extraction de données. Si les actions ne sont pas indentées, elles ne s'effectueront qu'une seule fois et contiendront les données de la dernière extraction de données. Pour l'exemple ci-dessus, la cinquième ligne du CSV sera imprimée, mais pas les quatre premières lignes. Avec des sous-zones, il faut indenter l'action dans le bon sous-groupe.

5.2 Filtre De Texte Structuré

5.2.1 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 une des méthodes suivantes.

- **Les champs sont séparés par un caractère.** Généralement les séparateurs sont une virgule ou un point-virgule. Le fichier CSV (valeurs séparées par virgule ou point virgule) est un exemple typique.
- **Les champs contiennent un nombre fixe de caractères**. En d'autres termes, les champs sont définis par des colonnes de largeur fixe.

Pour des exemples de données en 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.

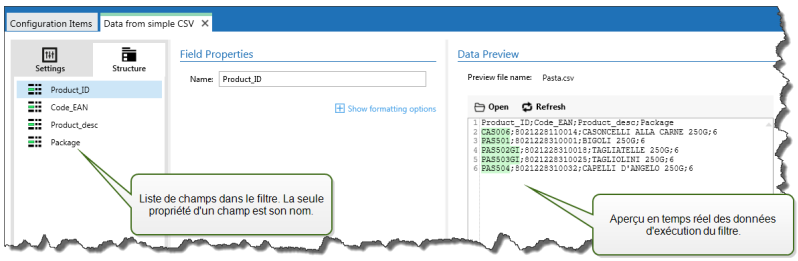
- **Importation de la structure en utilisant l'Assistant de fichier texte.** Dans ce cas, cliquer sur le bouton **Importer la Structure des Données** dans le ruban et suivre les instructions à l'écran. Quand l'Assistant est terminé, le type de base de données texte et tous les champs seront définis. Si la première ligne de données contient les noms de champs, l'Assistant peut les importer. C'est la méthode recommandée si le déclencheur reçoit toujours des données de structure identique.
- **Définition manuelle des champs.** 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, consulter l'article Définition des champs.
- **Lecture dynamique des champs.** Dans ce cas, le déclencheur peut recevoir des données de structures différentes, tel que des nouveaux noms de champs, cependant il ne faut pas mettre à jour le filtre pour chaque changement de structure. Cette méthode dynamique va automatiquement lire tous les champs dans le fichier de données et va les relier automatiquement aux variables ayant le même nom, même s'il existe des nouveaux champs ou s'il manque d'anciens champs,. Pour plus d'informations, voir l'article Activer la Structure Dynamique.

La section Aperçu de Données simplifie la configuration. Le résultat des filtres définis est surligné dans la zone d'aperçu à chaque changement de configuration. Cela permet de visualiser les données extraites pour chaque règle.

5.2.2 Définition Des Champs

La définition des champs est très facile pour les fichier texte structurés. Il y a deux options.

- **La délimitation définit les champs.** Dans ce cas, il y a des séparateurs, tels que virgule ou point virgule entre les champs. Il suffit définir les noms de champs dans le même ordre qu'ils apparaîtront dans les données reçues par un déclencheur.
- **Champs de largeur fixe.** Dans ce cas, il suffit de définir les noms de champs dans le même ordre qu'ils apparaîtront dans les données reçues par un déclencheur et de 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. Quand l'élément défini est

sélectionné, l'aperçu va surligner 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 sauvegardé.
- **Ouvrir.** Sélectionne un autre fichier sur lequel les règles du filtre vont s'appliquer.
- **Rafraîchir.** Relance les règles du filtre en fonction du contenu du nom du fichier d'aperçu. La section Aperçu de Données sera mise à jour 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. Les fonctions s'appliqueront dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début.** Enlève tous les espaces (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin.** Enlève tous les espaces (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Supprimer le caractère d'ouverture et 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}}` sera 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. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classique. Pour plus d'informations, consulter la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par des espaces.** Remplace tous les caractères de contrôle de la chaîne de caractères par des espaces (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.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Chariot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter l'article Introduire des caractères spéciaux

(Codes de Contrôle)

Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

EXEMPLE: Quand il reçoit les données "<CR><LF>", NiceLabel Automation 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.** 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.

5.2.3 Activer La Structure Dynamique

Le filtre de Texte Structuré à la possibilité d'identifier automatiquement dans les données les champs et leurs valeurs, ce qui élimine la nécessité de relier la *variable au champ*.

Cette fonctionnalité est utile quand le déclencheur reçoit la donnée de la structure changeable. La structure principale de données est identique, par ex. les champs sont délimités par une virgule, ou la même structure XML, mais **l'ordre** dans lequel les champs sont représentés est changé et/ou **le nombre de champs** a changé; il peut y avoir de nouveaux champs, ou certains champs ne sont plus disponibles. La structure sera automatiquement identifiée par le filtre. En même temps, les noms et valeurs de champs (paires `nom:valeur`) seront lues des données, éliminant la nécessité de mapper les variables manuellement.

L'action Utiliser un filtre de données ne propose pas la possibilité de mappage, car le mappage se fera de façon dynamique. Il n'y a même pas besoin de définir les variables d'étiquettes dans la configuration du déclencheur. L'action assignera 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 Impression de 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 ne sera signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Les variables manquantes sont ignorées en silence.

Configuration de la structure dynamique

Pour configurer la structure dynamique, activer l'option **Structure Dynamique** dans les propriétés du Filtre de Textes Structurés.

- 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.
- Et les données formatées, 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. Les fonctions s'appliqueront dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début.** Enlève tous les espaces (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin.** Enlève tous les espaces (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Supprimer le caractère d'ouverture et 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}}` sera 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. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classique. Pour plus d'informations, consulter la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par des espaces.** Remplace tous les caractères de contrôle de la chaîne de caractères par des espaces (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.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier,

tels que Retour Chariot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter l'article Introduire des caractères spéciaux (Codes de Contrôle)

Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

EXEMPLE: Quand il reçoit les données "<CR><LF>", NiceLabel Automation 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.** 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.

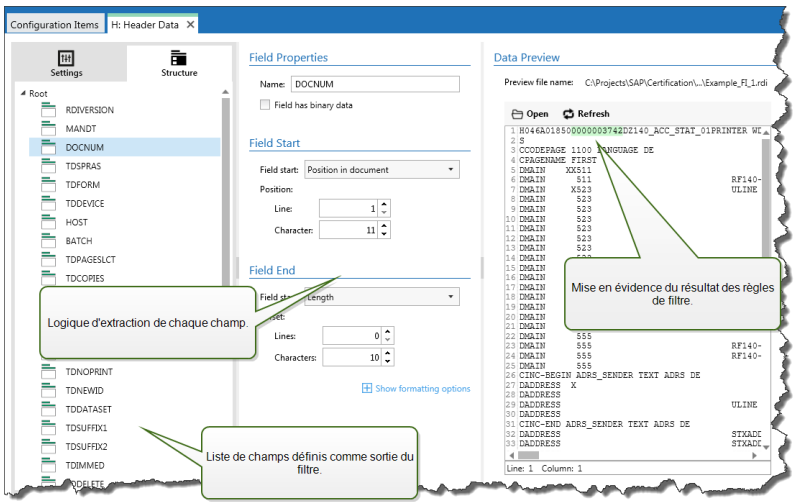
5.3 Filtre De Données Non-structurées

5.3.1 Filtre De Données Non-structurées

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

Ces données peuvent se présenter sous forme de documents exportés ou de rapports provenant d'un vieux système, de communication interceptée entre deux périphériques, d'une capture d'un flux d'impression etc. Le filtre permet d'extraire les champs individuels, les champs répétables dans les sous-zones, et même les paires `nom-valeur`.

Pour des exemples de données en texte structuré, voir les articles Données existantes, CSV Composé et Fichiers Binaires.



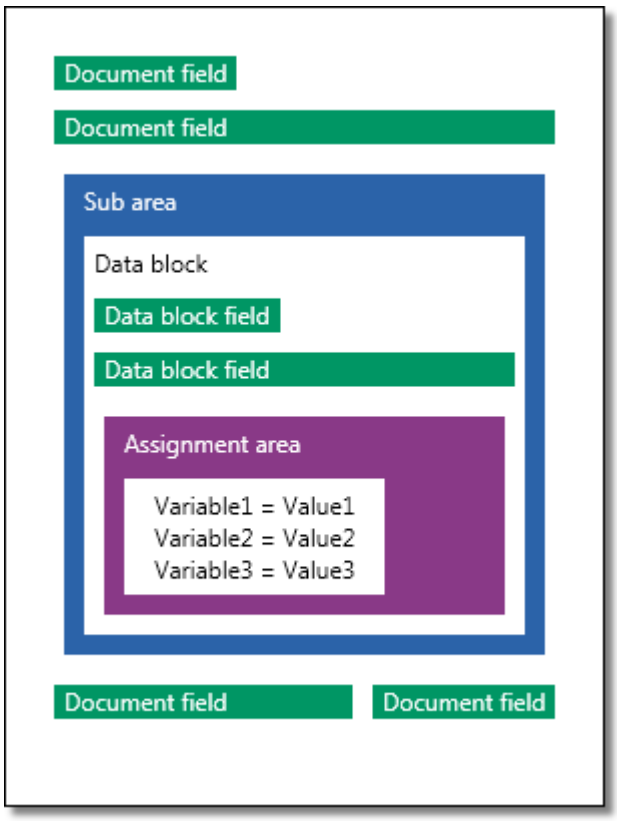
Définition de la structure

Eléments 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, consulter l'article 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, un nouveau niveau d'espace réservé sera défini dans "Utiliser le Filtre de Données", permettant d'associer les variables aux champs de ce niveau. Pour plus d'informations, consulter l'article Définition de sous-zones.
- **Zone d'affectation**. Spécifie l'emplacement des données répétables contenant les paires `nom-valeur`. Les noms de champs et leurs valeurs sont lus simultanément. Le mappage aux variables est fait automatiquement. Utiliser cette fonctionnalité pour adapter le filtre aux données d'entrée changeables, éliminant le temps de maintenance. La zone d'affectation peut être définie au niveau racine du document, ou dans la sous-zone. Pour plus d'informations, consulter l'article 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. Cela permet de 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 les blocs de données. Les paires `nom-valeur` peuvent être définies dans la zone d'affectation.



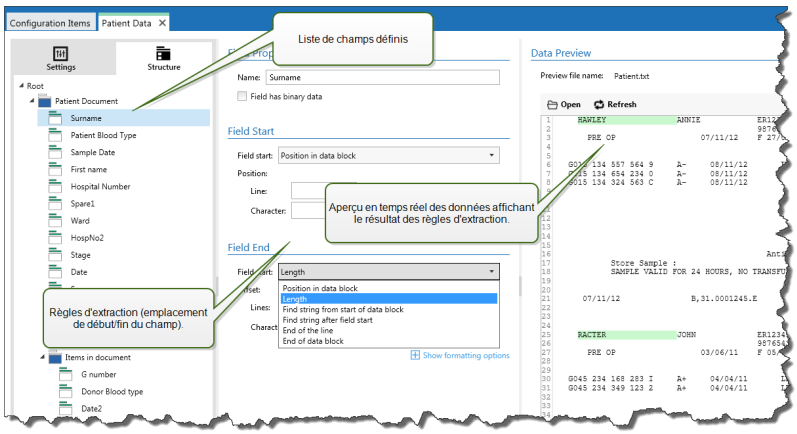
Général

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

- **Nom.** Spécifie le nom du filtre. Utiliser un nom descriptif qui identifiera ce que fait le filtre. Il est modifiable à tout moment.
- **Description.** Permet de décrire la fonctionnalité de ce filtre. L'utiliser pour écrire une courte description de la fonction du filtre.
- **Encodage.** Spécifie l'encodage des données avec lesquelles ce filtre va travailler.
- **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.

5.3.2 Définition Des Champs

Pour définir un champ, il faut définir son nom et une règle d'extraction de la valeur du champ des données. Quand le filtre sera exécuté, les règles d'extraction s'appliqueront aux données d'entrée et les résultat seront assignés aux champs.



Propriétés des champs

- **Nom.** Spécifie le nom de champ unique.
- **Champ contenant des données binaires.** Spécifie que le champ contiendra des données binaires. Ne l'activer que si les données attendues sont binaires.

Début de champ

- **Position dans le document.** Le point de début/fin est déterminé par la position codée en dur dans les données. 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.
- **Trouver une chaîne de caractères depuis le début du document.** Le point de début/-fin est défini par la position de recherche de chaîne de caractères. Quand la chaîne de caractères désirée est trouvée, 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 depuis une 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 depuis la chaîne de caractères.** Spécifie le décalage positif ou négatif après la chaîne de caractères recherchée.

    EXEMPLE: Le décalage défini sera inclus dans la recherche de chaîne de caractères avec les données extraites.

Fin de champ

- **Position dans le document.** Le point de début/fin est déterminé par la position codée en dur dans les données. 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.
- **Trouver une chaîne de caractères depuis le début du document.** Le point de début/-fin est défini par la position de recherche de chaîne de caractères. Quand la chaîne de caractères désirée est trouvée, 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 depuis une 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 depuis la chaîne de caractères.** Spécifie le décalage positif ou négatif après la chaîne de caractères recherchée.

      EXEMPLE: Le décalage défini sera inclus dans la recherche de chaîne de caractères avec les données extraites.
- **Trouver une chaîne de caractères 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 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. Les fonctions s'appliqueront dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début.** Enlève tous les espaces (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin.** Enlève tous les espaces (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Supprimer le caractère d'ouverture et 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}}` sera 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. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classique. Pour plus d'informations, consulter la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par des espaces.** Remplace tous les caractères de contrôle de la chaîne de caractères par des espaces (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.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Chariot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter l'article Introduire des caractères spéciaux (Codes de Contrôle)

  Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

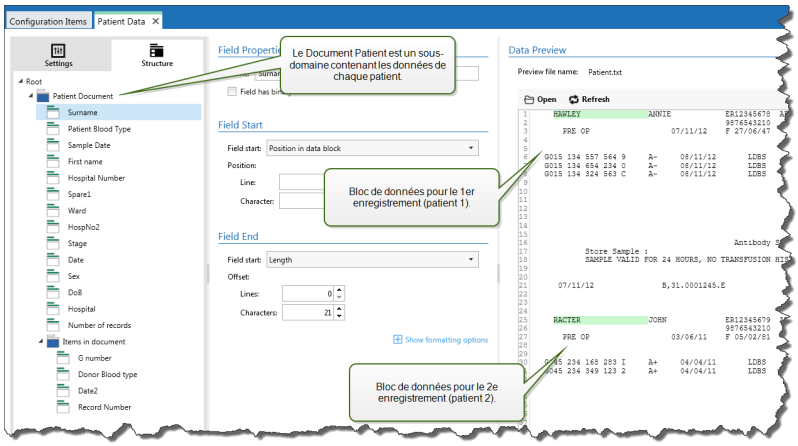
  EXEMPLE: Quand il reçoit les données "<CR><LF>", NiceLabel Automation les utilise comme une chaîne complète de 8 caractères. Il faut activer cette nouvelle option pour interpréter et utiliser les données comme deux caractères binaires `CR` (Retour Chariot- code ASCII 13) et `LF` (Passage à la Ligne - code ASCII 10).

- **Rechercher et supprimer tout avant.** 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.

5.3.3 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 affichera les sous-zones avec les emplacements indentés. Toute action indentée sous une telle sous-zone s'exécutera 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 de caractères.** 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.
    - **Le Bloc se termine par une chaîne de caractères.** Spécifie que les blocs de données se terminent par la chaîne de caractères fournie. Tout le contenu entre deux chaînes de caractères 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 de caractères.** Spécifie que les blocs de données sont séparés par la chaîne de caractères fournie. Tout le contenu entre deux chaînes de caractères appartient à un bloc de données distinct.

- **Début du Premier Bloc de Données.** Spécifie le point de départ du premier bloc de données et donc 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, consulter l'article Définition des champs.
- **Fin du Dernier Bloc de Données.** Spécifie la position de fin du dernier bloc de données et donc 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, consulter l'article 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, consulter l'article 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. Quand l'élément défini est sélectionné, l'aperçu va surligner 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 sauvegardé.
- **Ouvrir.** Sélectionne un autre fichier sur lequel les règles du filtre vont s'appliquer.
- **Rafraîchir.** Relance les règles du filtre en fonction du contenu du nom du fichier d'aperçu. La section Aperçu de Données sera mise à jour avec le résultat.

5.3.4 Définition Des Zones D'affectation

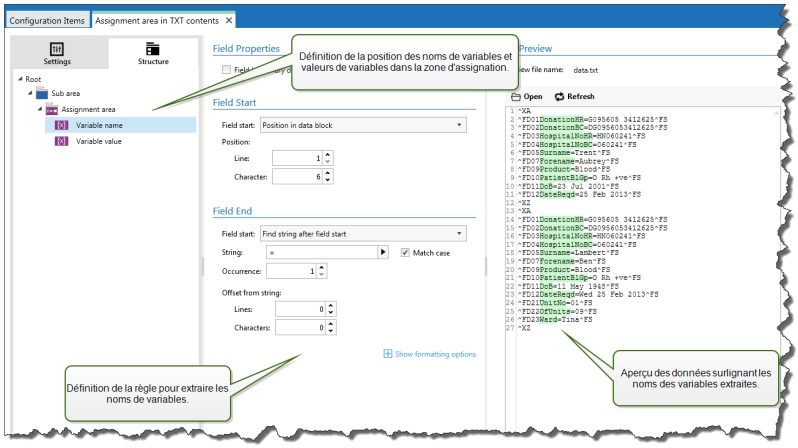
Le filtre de texte non-structuré a la capacité d'identifier automatiquement les champs et leurs valeurs dans les données, ce qui élimine la nécessité de relier la *variable au champ*.

Cette fonctionnalité est utile quand le déclencheur reçoit la donnée de la structure changeable. La structure principale de données est identique, par ex. les champs sont délimités par une virgule, ou la même structure XML, mais **l'ordre** dans lequel les champs sont représentés est changé et/ou **le nombre de champs** a changé; il peut y avoir de nouveaux champs, ou certains champs ne sont plus disponibles. La structure sera automatiquement identifiée par le filtre. En même temps, les noms et valeurs de champs (paires `nom:valeur`) seront lues des données, éliminant la nécessité de mapper les variables manuellement.

L'action Utiliser un filtre de données ne propose pas la possibilité de mappage, car le mappage se fera de façon dynamique. Il n'y a même pas besoin de définir les variables d'étiquettes dans la configuration du déclencheur. L'action assignera 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 Impression de 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 ne sera signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Les variables manquantes sont ignorées en silence.



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, consulter l'article Définition de 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

Lors de la création de la zone d'affectation, le filtre définira automatiquement deux espaces réservés, qui définiront les paires `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, consulter l'article Définition des champs.
- **Valeur de variable.** Spécifie le champ dont le contenu sera le nom de variable (composant `valeur` en paire). Configurer le champ en utilisant la même procédure que pour les champs de document. Pour plus d'informations, consulter l'article Définition des champs.

Exemple

La zone entre ^XA et ^XZ est la zone d'affectation. Chaque ligne de la zone l'affectation fournit les paires `nom-valeur`. Le nom est défini comme une valeur entre le 6ème 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=DG095605341262 5^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
```

Plus plus d'informations, consulter l'article Exemples.

5.4 Configuration Du Filtre XML

5.4.1 Filtre XML

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

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

Utiliser ce filtre chaque fois que le déclencheur reçoit des données encodées XML. Le filtre 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).

Chacun peut définir la structure d'un fichier XML, mais il est conseillé de l'importer du modèle de fichier XML donné. Cliquer sur le bouton **Importer la Structure de Données** dans le ruban. Dans la structure XML importée, la section Aperçu des Données affiche le contenu XML et met en évidence les éléments et attributs définis comme champs de sortie.

Pour des exemples de données XML, consulter l'article Données XML.

Définition de la Structure

Pour utiliser les éléments XML, il faut 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, consulter l'article Définition des Champs XML.
    - **Elément facultatif** Spécifie que cet élément n'est pas obligatoire. Ceci correspond à l'attribut `minOccurs=0` dans le schéma XML (fichier XSD). La variable associée 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 relier les champs définis aux variables

respectives dans l'action Utiliser le Filtre de Données. Pour plus d'informations, consulter l'article Définition des 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 fait automatiquement. Utiliser cette méthode pour adapter le filtre aux données d'entrée variables pour éliminer le temps de maintenance. Pour plus d'informations, consulter l'article Définition de la zone d'affectation XML.

La section Aperçu de Données simplifie la configuration. Le résultat d'un filtre défini sera 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.

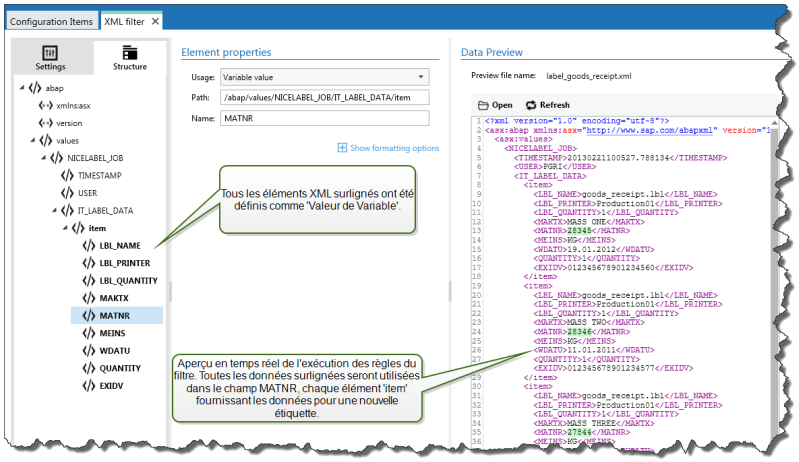
5.4.2 Définition Des Champs XML

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Quand un champ XML est défini, la valeur de l'élément sélectionné devient un champ. La définition du filtre va fournir ce champ et le relier à la variable dans une action. On peut extraire la valeur de l'élément ou la valeur de l'attribut.

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 dans la liste de structure s'affichera en caractères gras, indiquant qu'il est utilisé.
4. L'élément ou nom d'attribut sera utilisé comme le nom du champ de sortie.
5. La section Aperçu de Données surlignera 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. Les fonctions s'appliqueront dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début.** Enlève tous les espaces (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin.** Enlève tous les espaces (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Supprimer le caractère d'ouverture et 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}}` sera 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. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classique. Pour plus d'informations, consulter la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par des espaces.** Remplace tous les caractères de contrôle de la chaîne de caractères par des espaces (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.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Chariot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter l'article Introduire des caractères spéciaux (Codes de Contrôle)

  Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

  EXEMPLE: Quand il reçoit les données "<CR><LF>", NiceLabel Automation 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.** 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.

Aperçu des données

Cette section fournit un aperçu de la définition du champ. Quand l'élément défini est sélectionné, l'aperçu va surligner 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 sauvegardé.
- **Ouvrir.** Sélectionne un autre fichier sur lequel les règles du filtre vont s'appliquer.
- **Rafraîchir.** Relance les règles du filtre en fonction du contenu du nom du fichier d'aperçu. La section Aperçu de Données sera mise à jour avec le résultat.

5.4.3 Définition Des Éléments Répétables

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

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

Exemple

L'élément `<item>` est défini comme **Bloc de Données** et **Élément Répétable.** Ceci indique au filtre d'extraire toutes les occurrences de l'élément `<item>`, pas seulement le premier. Dans ce cas, l'élément **<item>** sera défini comme le sous-niveau dans l'action **Utiliser le Filtre de Données**. Il faut indenter les actions "Ouvrir l'Étiquette" et "Imprimer l'Étiquette" sous cet espace de sous-niveau, pour qu'elles soient bouclées autant de fois qu'il y a d'occurrences de l'élément `<item>`. Dans ce cas, trois fois.

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
<asx:values>
<NICELABEL_JOB>
<TIMESTAMP>20130221100527.788134</TIMESTAMP>
<USER>PGRI</USER>
<IT_LABEL_DATA>

<item>
<LBL_NAME>goods_receipt.lbl</LBL_NAME>
 <LBL_PRINTER>Production01</LBL_PRINTER>
```

```
 <LBL_QUANTITY>1</LBL_QUANTITY>
 <MAKTX>MASS ONE</MAKTX>
 <MATNR>28345</MATNR>
 <MEINS>KG</MEINS>
 <WDATU>19.01.2012</WDATU>
 <QUANTITY>1</QUANTITY>
 <EXIDV>012345678901234560</EXIDV>
</item>

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

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

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

5.4.4 Définition De La Zone D'affectation XML

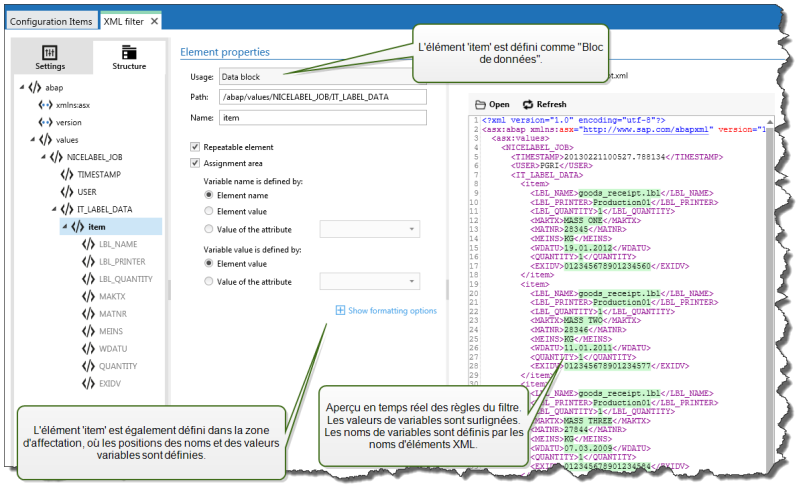
CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Le filtre XML a la possibilité d'identifier automatiquement les champs et leurs valeurs, ce qui élimine la nécessité de relier manuellement *variable et champ*.

Cette fonctionnalité est utile quand le déclencheur reçoit la donnée de la structure changeable. La structure principale de données est identique, par ex. les champs sont délimités par une virgule, ou la même structure XML, mais **l'ordre** dans lequel les champs sont représentés est changé et/ou **le nombre de champs** a changé; il peut y avoir de nouveaux champs, ou certains champs ne sont plus disponibles. La structure sera automatiquement identifiée par le filtre. En même temps, les noms et valeurs de champs (paires `nom:valeur`) seront lues des données, éliminant la nécessité de mapper les variables manuellement.

L'action Utiliser un filtre de données ne propose pas la possibilité de mappage, car le mappage se fera de façon dynamique. Il n'y a même pas besoin de définir les variables d'étiquettes dans la configuration du déclencheur. L'action assignera 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 Impression de 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 ne sera signalée si le champ disponible dans les données d'entrée n'a pas de variable d'étiquette correspondant. Les variables manquantes sont ignorées en silence.



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. Il faut dire 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. La variable d'étiquette doit avoir le même nom pour permettre le fonctionnement du mappage automatique.
- **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.

ATTENTION : 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. Les

fonctions s'appliqueront dans l'ordre sélectionné dans l'interface utilisateur, de haut en bas.

- **Supprimer les espaces au début.** Enlève tous les espaces (code décimal ASCII 32) du début de la chaîne de caractères.
- **Supprimer les espaces à la fin.** Enlève tous les espaces (code décimal ASCII 32) à la fin de la chaîne de caractères.
- **Supprimer le caractère d'ouverture et 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}}` sera 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. NiceLabel Automation utilise la syntaxe .NET Framework pour les expressions classique. Pour plus d'informations, consulter la Base de Connaissances article KB250.

- **Remplacer les caractères non-imprimables par des espaces.** Remplace tous les caractères de contrôle de la chaîne de caractères par des espaces (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.** Efface tous les caractères dans la chaîne de caractères. Les caractères non-imprimables sont des caractères ayant une valeur ASCII décimale comprise entre 0-31 et 127-159.
- **Décoder les caractères spéciaux.** Les caractères spéciaux (ou caractères de contrôle) sont des caractères qui ne sont pas disponibles sur le clavier, tels que Retour Chariot et Passage à la Ligne. NiceLabel Automation utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot et <LF> pour Passage à la Ligne. Pour plus d'informations, consulter l'article Introduire des caractères spéciaux (Codes de Contrôle)

  Cette option convertit les caractères spéciaux de la syntaxe NiceLabel en caractères binaires réels.

  EXEMPLE: Quand il reçoit les données "<CR><LF>", NiceLabel Automation 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.** 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.

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>
```

Plus plus d'informations, consulter l'article Exemples.

5.5 Paramétrer Le Nom De L'étiquette Et De L'imprimante Dans Les Données Entrées

En principe, les filtres sont utilisés pour extraire les valeurs des données reçues et les envoyer aux variables de l'étiquette pour l'impression. Dans ce cas, le nom de l'étiquette ou de l'imprimante est codé en dur dans les actions. Par exemple, l'action Ouvrir l'Étiquette va coder en dur le nom de l'étiquette, et l'action Définir l'Imprimante va coder en dur le nom de l'imprimante. Toutefois, les données d'entrée peuvent également fournir les *méta-données*, valeurs utilisées par NiceLabel Automation, mais 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.** Il faut définir de nouveaux champs pour que les données entrée comportent aussi les champs de méta-données.
2. **Définition de Variable.** il faut définir manuellement les variables sur lesquelles les méta-données seront stockées; elles n'existent pas sur l'étiquette et ne peuvent pas être importées. Utiliser des noms intuitifs, tels que `NomEtiquette`, `NomImprimante`, et `Quantité`. N'importe quel nom de variable conviendra.

3. **Reconfiguration du mappage.** Il faut configurer manuellement l'action Utiliser le Filtre de Données pour relier les méta-champs aux variables.
4. **Reconfiguration d'action.** Il faut reconfigurer l'action **Ouvrir l'Étiquette** pour ouvrir l'étiquette spécifiée par la variable `NomEtiquette`, 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 d'étiquette, le nom d'imprimante et la quantité d'étiquettes. Le filtre de Texte Structuré va extraire tous les champs, envoyer les valeurs relatives aux variables d'étiquette et utiliser les *méta-données* pour configurer les actions "Ouvrir 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
```

Pour plus d'informations, consulter l'article Exemples.

6 Configuration des déclencheurs

6.1 Comprendre Les Déclencheurs

CONSEIL: Les fonctionnalités de cet élément ne sont pas disponibles dans tous les produits NiceLabel Automation.

NiceLabel Automation est une application basée sur des événements qui va déclencher l'exécution d'actions lors d'un changement de l'événement surveillé. 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. La tâche principale du déclencheur est de reconnaître le changement dans l'événement, récupérer les données fournies par l'événement et ensuite effectuer les actions. La majorité des déclencheurs sont conçus pour écouter passivement l'apparition de l'événement surveillé, mais 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 reliées. Dès que les données sont enregistrées dans les variables, des actions vont pouvoir utiliser les variables. ex: l'action Imprimer l'étiquette.

Quand l'événement apparaît, la donnée d'entrée est fournie et enregistrée dans un fichier temporaire sur le disque, dans le dossier de service de l'utilisateur `%temp`. La variable interne `DataFileName` fait référence à l'emplacement du fichier temporaire. Le fichier est effacé quand le déclencheur termine son exécution.

Propriétés du déclencheur

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

- **Paramètres.** Définit les paramètres principaux du déclencheur. Définir l'événement qui sera surveillé pour activer le déclencheur, ou définir le canal de communication entrante.

Les paramètres comprennent la sélection du 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** Cette section définit les variables nécessaires dans le déclencheur. Généralement les variables sont importées du masque d'étiquette; il faut donc les relier 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, consulter l'article Actions.

Types de déclencheurs

- **Déclencheur 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 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 en utilisant les instructions `INSERT`, `UPDATE` et `INSERT SQL`.
- **Déclencheur Serveur TCP/IP.** Surveille le flux de données brutes entrant sur le socket 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'information.
- **Déclencheur Serveur HTTP.** Surveille le flux de données au format HTTP arrivant sur le socket défini. Le contenu du flux entrant peut être analysé dans des filtres et utilisé dans des actions. L'authentification d'utilisateur peut être activée. Il peut être bidirectionnel, fournissant un retour d'information.
- **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'information.

Traitement des Erreurs dans les Déclencheurs

- **Erreurs de Configuration.** Le déclencheur est en état d'erreur tant qu'il n'est pas configuré convenablement ou complètement. 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. Une erreur au niveau inférieur de la configuration

se propagera jusqu'au niveau le plus haut, ce qui rendra la localisation de l'erreur plus facile.

> **EXEMPLE:** Par exemple, avec une action en état d'erreur, toutes les actions de niveau supérieur indiqueront la situation d'erreur, l'icône d'erreur s'affichera dans l'onglet Action et dans le nom du déclencheur.

- **Chevauchements de Configurations.** 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'information sur le travail d'impression

Pour plus d'informations, consulter l'article Retour d'information sur le travail d'impression.

6.2 Définition Des Déclencheurs

6.2.1 Déclencheur Fichier

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

L'événement Déclencheur fichier survient quand un fichier ou un ensemble de fichiers dans un dossier surveillé change. L'apparition d'un nouveau fichier active aussi un déclencheur. 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 a un nouvel horodatage.

Utilisation typique : Le système utilisé exécute une transaction qui va générer un déclencheur fichier dans le dossier partagé. Le contenu de données peut être structuré en format CSV, XML et autres formats, 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 de renseignements concernant l'analyse et l'extraction de données, consulter l'article Comprendre les Filtres.

Général

Cette section permet de configurer les paramètres des déclencheurs fichiers les plus importants.

- **Nom.** Spécifie le nom du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder et pour les exécuter ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. L'utiliser pour écrire une courte description de la fonction du déclencheur.

- **Détecter le fichier spécifié.** Spécifie le chemin et le nom du fichier à surveiller.
- **Détecter un ensemble de fichiers dans le dossier spécifié.** Spécifie le chemin vers le dossier où surveiller les changements de fichiers et les noms de fichiers. Utiliser les caractères génériques standards 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 réseau partagées.

- **Détecter automatiquement les changements.** L'application répondra aux changements du fichier dès que le fichier sera créé ou changé. 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.
- **Vérifier les changements dans le dossier toutes les ... (millisecondes).** L'application va scanner le dossier pour rechercher des changements de fichiers à chaque intervalle de temps défini. 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.** Spécifie l'ouverture du fichier déclencheur en mode exclusif. Aucune autre application ne peut accéder au fichier en même temps. C'est la sélection par défaut.
- **Ouvrir le fichier en lecture seule.** Spécifie l'ouverture du fichier déclencheur en mode lecture seule.
- **Ouvrir le fichier en lecture et écriture.** Spécifie l'ouverture du fichier de déclenchement en mode lecture/écriture.
- **Période de tentative d'ouverture du fichier.** Spécifie la période de temps durant laquelle NiceLabel Automation va essayer d'ouvrir le fichier déclencheur. Si l'accès au fichier n'est pas possible au bout de cette période de temps, NiceLabel Automation va rapporter une erreur.

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

- **Vérifier 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 peuvent passer inaperçus, donc le changement de longueur du fichier peut aider à

déclencher les actions.

- **Ignorer les fichiers déclencheurs vides.** Si le fichier de déclenchement est vide, il sera ignoré. Les actions ne seront pas exécutées.
- **Effacer le fichier déclencheur.** Après détection du changement dans le fichier déclencheur et activation du déclencheur, le fichier sera effacé. L'activation de cette option permet de nettoyer le dossier des fichiers déjà traités.

  NOTE: NiceLabel Automation crée toujours une sauvegarde de la donnée de déclenchement reçue; ici le contenu du fichier déclencheur, et l'enregistre dans un fichier à nom 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 déclenchement sauvegardées est référencée par la variable interne *DataFileName*.

- **Vider le contenu du fichier.** L'exécution des actions vide le contenu du fichier déclencheur. C'est utile quand des applications tierces joignent des données au fichier déclencheur. Il faut conserver le fichier pour y joindre les données, sans pour autant imprimer les anciennes données.
- **Suivre les changements quand le déclencheur est inactif.** Spécifier s'il faut activer le déclencheur avec les fichiers qui ont changé quand le déclencheur n'était pas activé. 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 quand le serveur est arrêté. Quand NiceLabel Automation est à nouveau en ligne, les fichiers déclencheurs existants peuvent être traités.

Autre

Les options de la section **Commentaires du moteur d'impression** précisent la communication avec le moteur d'impression.

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression supervisée.** Active le mode d'impression synchronisée. 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, NiceLabel Automation va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement d'une valeur trop longue dans la variable d'étiquette, ou de paramétrage d'une valeur dans une variable inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs dépassant la longueur de la variable définie dans l'éditeur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs 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 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** En imprimant avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables spécifiées dans le fichier de commande (utilisant la commande SET), mais non définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue lorsqu'une zone d'affectation est définie dans le filtre pour extraire toutes les paires *nom:valeur*, mais qu'il y a moins de variables définies dans l'étiquette.

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

- **Langage de Script.** Spécifie le langage du script activé 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 dans un 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 dans un fichier.** Activer cette option pour enregistrer les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

ATTENTION : Il faut activer l'impression supervisée, sinon NiceLabel Automation ne sera pas en mesure de détecter l'erreur durant l'exécution. Pour plus d'informations, Consulter l'article Mode d'impression Synchrone.

NOTE: NiceLabel Automation enregistre toujours les données reçues dans un fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

6.2.2 Déclencheur Port Série

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

L'événement du déclencheur 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. A 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 de renseignements concernant l'analyse et l'extraction de données, consulter l'article Comprendre les Filtres.

Général

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

- **Nom.** Spécifie le nom du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder et pour les exécuter ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. L'utiliser pour écrire une courte description de la fonction du déclencheur.
- **Port.** Spécifie le numéro de port série (COM) sur lequel les données entrantes seront acceptées. 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 ne sera pas effectuée au 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 l'article Introduire des caractères spéciaux (Codes de Contrôle).
- **Utiliser l'interrogation de données.** Spécifie que le déclencheur questionnera 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 l'article Introduire des caractères spéciaux (Codes de Contrôle).

Autre

Les options de la section **Commentaires du moteur d'impression** précisent la communication avec le moteur d'impression.

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression supervisée.** Active le mode d'impression synchronisée. 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, NiceLabel Automation va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement d'une valeur trop longue dans la variable d'étiquette, ou de paramétrage d'une valeur dans une variable inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs dépassant la longueur de la variable définie dans l'éditeur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs 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 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** En imprimant avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables spécifiées dans le fichier de commande (utilisant la commande SET), mais non définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue lorsqu'une zone d'affectation est définie dans le filtre pour extraire toutes les paires *nom:valeur*, mais qu'il y a moins de variables définies dans l'étiquette.

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

- **Langage de Script.** Spécifie le langage du script activé 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 dans un 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 dans un fichier.** Activer cette option pour enregistrer les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

  ATTENTION : Il faut activer l'impression supervisée, sinon NiceLabel Automation ne sera pas en mesure de détecter l'erreur durant l'exécution. Pour plus d'informations, Consulter l'article Mode d'impression Synchrone.

  NOTE: NiceLabel Automation enregistre toujours les données reçues dans un fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

6.2.3 Déclencheur De Base De Données

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

L'événement déclencheur de la base de données survient quand un changement est détecté dans la base de données surveillée. Il peut s'agir de nouvelles données ou de données existantes qui ont été mises à jour. Le déclencheur de la 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 : Le 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 données mises à jour ou les nouvelles données et va imprimer leurs contenus sur les étiquettes.

Général

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

- **Nom.** Spécifie le nom du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder et pour les exécuter ensuite dans Automation Manager.

- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. L'utiliser pour écrire une courte description de la fonction du déclencheur.
- **Connexion à la base de données.** Spécifie la chaîne de connexion à la base de données. L'interface de la base de données s'ouvre en cliquant sur le bouton **Définir.** Elle permet de configurer une connexion à la base de données avec son type, le nom de la table et les détails de l'utilisateur. Il faut pour cela se connecter à la base de données avec des 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 (fichiers séparés par virgule) et feuilles de calcul Microsoft Excel.

  NOTE: Les détails de configuration dépendent du type de base de données sélectionné. Les options de la boite 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 connectivité de la base de données, consulter l'article Accéder aux bases de données.
- **Vérification de la base de données à intervalles de temps.** Spécifie l'intervalle de temps pour sonder les enregistrements de la base de données.
- **Options de détection et options 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'objet Pour Chaque Enregistrement, dans lequel on peut relier les champs des tables aux variables des étiquettes.

  Collecte d'enregistrements basée sur une valeur incrémentale unique de champ

  Dans ce cas, le déclencheur surveille un champ numérique spécifique auto-incrémental de 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 seront collectées. Pour configurer cette option, il faut sélectionner le nom de la table où les données 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 fenêtre de dialogue. La configuration et/ou démarrer/arrêter ce déclencheur peut être chargée en toute sécurité dans l'Automation Manager tout en conservant la dernière valeur mémorisée. Toutefois, si vous enlevez la configuration de l'Automation Manager et la remettez, la valeur du dernier champ clé mémorisé sera réinitialisée à la `valeur par défaut du champ clé`.

  Collecter les enregistrements et les effacer

  Dans ce cas, tous les enregistrements de la table sont collectés puis ils sont effacés de la

table. Pour configurer cette option, il faut sélectionner le nom de la table où résident les enregistrements (`nom de table`) et spécifier la clé primaire dans la table (`champs clés`). Il peut arriver d'avoir un tableau sans clé primaire, il est toutefois fortement recommandé de définir une clé primaire. S'il existe une clé, les données sont effacées une par une au fur et à mesure du traitement des enregistrements dans les actions.

ATTENTION : Si la clé primaire n'existe pas, toutes les données collectées dans le déclencheur actuel sont effacées en même temps. C'est bien quand il n'y a pas d'erreurs de traitement des données. Mais si une erreur survient durant le traitement des données, l'Automation va arrêter le traitement de toutes les données. Comme toutes les données récupérées durant cet intervalle de sondage ont déjà été effacées sans avoir été traitées, cela peut causer une perte de données. C'est pourquoi c'est une bonne idée d'avoir une clé primaire.

Exemples de Code SQL

NOTE: Ces requêtes SQL sont en lecture seule et sont seulement fournies comme référence. Pour fournir les requêtes SQL personnalisée, sélectionner la méthode de détection **Trouver et gérer les enregistrements avec une requête SQL.**

Modèle de table

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 de 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 la première donnée, la valeur de `ID` est 1, pour la deuxième donnée 2, etc. L'utilisation de la variable est spécifié 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 contient l'index primaire.

```
DELETE FROM [Table]
```

Quand l'index primaire n'est pas défini dans la table, toutes les données sont effacées de la table quand la première donnée a été traitée.

Récupérer les enregistrements et les mettre à jour

Dans ce cas, tous les enregistrements sont collectés de la table et ensuite mis à jour. Une valeur personnalisée peut être placée dans un champ de la table pour indiquer que

'ces données ont déjà été imprimées'. Pour configurer cette option, il faut sélectionner le nom de la table où résident les données `(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`).

Il peut arriver d'avoir une table sans clé primaire, il est toutefois fortement recommandé de définir une clé primaire. S'il existe une clé , les données sont mises à jour une par une au fur et à mesure du traitement des enregistrements dans les actions.

ATTENTION : Si la clé primaire n'existe pas, toutes les données obtenues dans le déclencheur actuel seront mises à jour en même temps. C'est bien quand il n'y a pas d'erreurs de traitement des données. Mais si une erreur survient durant le traitement des données, l'Automation va arrêter le traitement de toutes les données. Comme toutes les données récupérées durant cet intervalle de sondage ont déjà été effacées sans avoir été traitées, cela peut causer une perte de données. C'est pourquoi c'est une bonne idée d'avoir une clé primaire.

Exemples de Code SQL

NOTE: Ces requêtes SQL sont en lecture seule et sont seulement fournies comme référence. Pour fournir les requêtes SQL personnalisée, sélectionner la méthode de détection **Trouver et gérer les enregistrements avec une requête SQL.**

Modèle de table

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.

```
UPDATE [Table]
SET [AlreadyPrinted] = Valeur de mise à jour
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 la première donnée, la valeur de `ID` est 1, pour la deuxième donnée 2, etc. L'utilisation de la variable est spécifié 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 la table n'a pas d'index primaire.

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

Quand l'index primaire n'est pas défini dans la table, toutes les données de la table sont mises à jour quand la première donnée a été traitée.

Trouver et gérer les données avec requêtes SQL personnalisée

Dans ce cas, les requêtes SQL pour l'extraction des enregistrements et les mises à jour des champs doivent être entièrement rédigées. Pour configurer cette option, il faut donner une requête SQL personnalisée pour trouver les données `(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

Modèle de table

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 SQL de Recherche (Search).

Pour trouver les données qui n'ont pas encore été imprimées, 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
 (sélectionner de la table) Où AlreadyPrinted <> 'Y' ou AlreadyPrinted est
NULL
```

Deux données vont être extraites de la table ci-dessus, avec les valeurs ID 2 et 3. La première donnée a déjà été imprimée et sera ignorée.

Exemple de requête SQL mise à jour (update).

Pour marquer les données déjà imprimées 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. 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 données traitées actuellement (la valeur du champ `ID` doit être la même que la valeur de la donnée actuelle). 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 un requête personnelle si l'option **Trouver et gérer les enregistrements avec une requête SQL** est sélectionnée**.**

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 SQL 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 est ouvert et si quelques variables sont ajoutées au script, cliquer sur le bouton **Test** deux fois (pour fermer et ouvrir la section Aperçu de Données) pour mettre à jour la liste de variables dans l'aperçu.

- **Simulation d'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 l'exécution des actions.** Spécifie que les enregistrements seront mis à jour avant que les actions définies pour ce déclencheur soient exécutées.
- **Après l'exécution des actions.** Spécifie que les enregistrements seront mis à jour après que les actions définies pour ce déclencheur soient exécutées. Généralement les données sont mises à jour après leur traitement.

NOTE: Mais si nécessaire, elles peuvent être mises à jour durant l'exécution des actions. Pour plus d'informations, se référer à :

Autre

Les options de la section **Commentaires du moteur d'impression** précisent la communication avec le moteur d'impression.

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression supervisée.** Active le mode d'impression synchronisée. 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, NiceLabel Automation va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement d'une valeur trop longue dans la variable d'étiquette, ou de paramétrage d'une valeur dans une variable inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs dépassant la longueur de la variable définie dans l'éditeur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs 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 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** En imprimant avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables spécifiées dans le fichier de commande (utilisant la commande SET), mais non définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue lorsqu'une zone d'affectation est définie dans le filtre pour extraire toutes les paires *nom:valeur*, mais qu'il y a moins de variables définies dans l'étiquette.

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

- **Langage de Script.** Spécifie le langage du script activé 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 dans un 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 dans un fichier.** Activer cette option pour enregistrer les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

ATTENTION : Il faut activer l'impression supervisée, sinon NiceLabel Automation ne sera pas en mesure de détecter l'erreur durant l'exécution. Pour plus d'informations, Consulter l'article Mode d'impression Synchrone.

NOTE: NiceLabel Automation enregistre toujours les données reçues dans un fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

6.2.4 Déclencheur Serveur TCP/IP

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

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

Utilisation typique : Le système existant exécute une transaction, qui en fait envoie les données au serveur NiceLabel Automation sur un socket spécifique. Le contenu de 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 de renseignements 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 du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder et pour les exécuter ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. L'utiliser pour écrire une courte description de la fonction du 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 la sécurité, consulter 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.

- **A la déconnexion du client.** Spécifie que le déclencheur va s'activer 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*.

- **Selon le nombre de caractères reçus.** Spécifie que le déclencheur va s'activer quand le nombre de caractères requis a été reçu. 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 va s'activer chaque fois qu'il reçoit la séquence de caractères requise. 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 le déclencheur ne sera pas coupée des données, mais sera inclus*e dans les données. Le déclencheur recevra le flux de données complet.
- **Lorsque rien n'est reçu après l'intervalle de temps spécifié.** Spécifie que le déclencheur va s'activer après un intervalle de temps requis suivant la réception du dernier caractère.

Exécution

- **Permettre 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 des 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.
- **Codage 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** précisent la communication avec le moteur d'impression.

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression supervisée.** Active le mode d'impression synchronisée. 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, NiceLabel Automation va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement d'une valeur trop longue dans la variable d'étiquette, ou de paramétrage d'une valeur dans une variable inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs dépassant la longueur de la variable définie dans l'éditeur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs 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 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** En imprimant avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables spécifiées dans le fichier de commande (utilisant la commande SET), mais non définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue lorsqu'une zone d'affectation est définie dans le filtre pour extraire toutes les paires *nom:valeur*, mais qu'il y a moins de variables définies dans l'étiquette.

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

- **Langage de Script.** Spécifie le langage du script activé 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 dans un 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 dans un fichier.** Activer cette option pour enregistrer les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

  ATTENTION : Il faut activer l'impression supervisée, sinon NiceLabel Automation ne sera pas en mesure de détecter l'erreur durant l'exécution. Pour plus d'informations, Consulter l'article Mode d'impression Synchrone.

  NOTE: NiceLabel Automation enregistre toujours les données reçues dans un fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

6.2.5 Déclencheur Serveur HTTP

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

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

L'événement déclencheur HTTP survient quand des données sont reçues sur le socket surveillé (numéro d'adresse IP et de port). Au contraire du déclencheur TCP/IP, les données reçues ne sont pas un flux de données brutes, elles doivent comporter l'entête HTTP normalisée. L'application tierce doit utiliser les méthodes de requête POST ou GET et procurer des données dans le corps du message ou dans la chaîne de requête. Le type de média Internet utilisé dans le corps du message n'est pas important (Type MIME, ou Type Content). NiceLabel Automation recevra le message. Définir un filtre permet d'extraire les données requises contenues dans le message.

Utilisation typique : Le système existant exécute une transaction, qui en fait envoie les données au serveur NiceLabel Automation formatées en message HTTP POST sur un socket spécifique. Le contenu de données peut être structuré en format CSV, XML et autres formats, 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 imprimer les valeurs extraites sur les étiquettes. Pour plus de renseignements concernant l'analyse et l'extraction de données, consulter l'article Comprendre les Filtres.

Fourniture de données

Pour fournir les données pour le déclencheur HTTP utiliser 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.

Une URL typique contenant une chaîne (string) de requête (query) se présente comme suit :

```
http://serveur/chemin/?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 habituellement composée d'une série de paires `nom:valeur`, dans laquelle chaque nom de champ est séparé de sa valeur par un signe égal (=). Les séries de paires sont séparées par le signe 'et commercial' (&). La chaîne de requête typique fournit les valeurs des champs (variables), comme suit :

```
champ1=valeur1&champ2=valeur2&champ3=valeur3
(field1=value1&field2=value2&field3=value3)
```

Le déclencheur HTTP est capable d'extraire les valeurs de tous les champs et de les enregistrer dans les variables de même nom, donc il ne faut pas définir de filtres pour extraire les valeurs 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. Si les variables n'existent pas dans l'étiquette, les valeurs sont ignorées et aucune erreur n'est rapportée.
- Il faut définir les variables du déclencheur quand leurs valeurs sont nécessaires pour certaines actions de ce 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 procurer le message dans le corps de la requête HTTP.

N'importe quelles données et structure de données peuvent être envoyées dans le corps, à condition de les traiter dans les filtres NiceLabel Automation. Le contenu peut être formaté en XML, CSV, il peut être sous forme de texte, il peut même être sous forme de 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 du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder et pour les exécuter ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. L'utiliser pour écrire une courte description de la fonction du 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. Dans des cas plus complexes, le contenu personnalisé peut être renvoyé à l'application procurant les données, soit dans une simple chaîne de retour, soit en données binaires. Par exemple: un aperçu de l'étiquette ou un flux d'impression.

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

```
http://serveur:port/chemin/?query_string
```

- **Serveur**. Ceci est l'adresse IP ou FQDN de la machine sur laquelle NiceLabel Automation est installé.
- **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 la sécurité, consulter 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 (\).

CONSEIL: Cette option est disponible dans NiceLabel Automation Enterprise.

- **Connexion Sécurisée (HTTPS).** Active la couche de transport sécurisée pour vos messages HTTP et évite l'écoute clandestine. Pour plus d'informations sur son paramétrage, consulter l'article Utilisation de la couche de transport sécurisée (HTTPS).
- **Chaîne de requête.** Spécifie les paires nom-valeur dans l'URL. Un paramètre optionnel, 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 répondra par le code 200, indiquant que les données ont bien été reçues, mais ceci ne donne pas d'informations concernant le succès des actions du déclencheur.

  Cette option spécifie que le déclencheur n'envoie pas la réponse immédiatement après la réception des données, mais attend que toutes les actions soient exécutées et envoie ensuite le code de réponse indiquant le succès de l'action exécutée. Quand 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 rapport concernant le processus d'impression, il faut activer le mode d'impression **synchrone**. Pour plus d'information, consulter l'article Impression en mode 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. Le nombre dépend aussi des performances physiques de votre serveur.

- **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. Les données de réponse seront envoyées à l'extérieur sous forme de rapport, formatées selon le type de média défini. L'option **Variable** active le type de média variable. dans ce cas, il faut sélectionner la variable qui contiendra le type de média.

  NOTE: Si le type de contenu n'est pas spécifié, NiceLabel Automation utilisera `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.

  NOTE: 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 ex.`image/jpeg` ou `application/octet-stream`.

- **Entêtes supplémentaires.** Permet de définir des entête 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 .

  CONSEIL: Dans les données et l'entête de la réponse, 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 entrant comprennent le nom d'utilisateur et mot de passe. Le déclencheur acceptera seulement les messages HTTP dont les informations d'identification sont correctes. Pour plus d'informations concernant la sécurité, consulter l'article Sécuriser l'accès aux déclencheurs.
- **Groupe 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. 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 dans le NiceLabel Control Center. Pour choisir le groupe qui va être autorisé à accéder au déclencheur 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 portant un nom spécifique doit exister sur NiceLabel Control Center quand le déclencheur démarre. N'importe quel nom de groupe peut être utilisé dans la configuration de Automation Builder. Mais il faut que le nom définit à la fin dans NiceLabel Control Center corresponde à celui qui est défini dans la configuration.

CONSEIL: Les utilisateurs s'authentifient avec leurs identifiants et mots de passe tels qu'ils sont définis dans **NiceLabel Control Center > Administration > Utilisateurs et Groupes**. 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** précisent la communication avec le moteur d'impression.

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression supervisée.** Active le mode d'impression synchronisée. 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, NiceLabel Automation va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement d'une valeur trop longue dans la variable d'étiquette, ou de paramétrage d'une valeur dans une variable inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs dépassant la longueur de la variable définie dans l'éditeur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs 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 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** En imprimant avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables spécifiées dans le fichier de commande (utilisant la commande SET), mais non définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue lorsqu'une zone

d'affectation est définie dans le filtre pour extraire toutes les paires *nom:valeur*, mais qu'il y a moins de variables définies dans l'étiquette.

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

- **Langage de Script.** Spécifie le langage du script activé 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 dans un 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 dans un fichier.** Activer cette option pour enregistrer les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

  ATTENTION : Il faut activer l'impression supervisée, sinon NiceLabel Automation ne sera pas en mesure de détecter l'erreur durant l'exécution. Pour plus d'informations, Consulter l'article Mode d'impression Synchrone.

  NOTE: NiceLabel Automation enregistre toujours les données reçues dans un fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

6.2.6 Déclencheur Web Service

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

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

Le déclencheur Web Service s'active quand les données arrivent sur le socket surveillé (numéro d'adresse IP et de port). 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 disponible pour chaque déclencheur Web Service défini.

Web Service peut fournir un rapport d'informations sur le travail d'impression, mais il faut activer le mode de traitement **synchrone**. Pour plus d'informations, consulter l'article Retour d'information sur le travail d'impression.

En général, Web Service est utilisé par des programmeurs pour intégrer l'impression d'étiquettes dans leurs applications. Le système existant exécute une transaction, qui en fait envoie les données au serveur NiceLabel Automation sur un socket spécifique, formatées comme un message SOAP. Le contenu de 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 de renseignements concernant l'analyse et l'extraction de données, consulter l'article Comprendre les Filtres.

Général

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

- **Nom.** Spécifie le nom du déclencheur. Les noms permettent de distinguer les différents déclencheurs lors de la configuration dans Automation Builder et pour les exécuter ensuite dans Automation Manager.
- **Description.** Procure la possibilité de décrire la fonctionnalité de ce déclencheur. L'utiliser pour écrire une courte description de la fonction du 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.

- **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 la sécurité, consulter 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 son paramétrage, consulter l'article Utilisation de la couche de transport sécurisée (HTTPS).
- **Nombre maximum d'appels concurrents.** 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 concernant l'état du travail d'impression créé. L'exécution de l'action peut être supervisée. Le déclencheur rapportera la réussite de chaque événement d'exécution. Pour activer le retour d'information sur le processus d'impression, il faut activer le Mode d'impression Synchrone.

Les méthodes (fonctions) suivantes sont exposées dans le déclencheur Web Service :

- **ExecuteTrigger.** - Exécuter le déclencheur Cette méthode accepte l'introduction de données dans le traitement et renvoie en option le rapport d'informations de statut. Un des paramètres d'entrée active ou désactive le retour d'informations. Si l'option est activé, le rapport d'information contiendra 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 durant la création du fichier d'impression. Si l'ID est supérieure à 0, une erreur est survenue durant le 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. Il est possible de récupérer toutes les données désirées dans l'une des structures existantes. il est possible d'utiliser des données binaires dans la réponse.
- **ExecuteTriggerAndSetVariables.** - Exécuter le déclencheur et définir les variables. Similaire à **ExecuteTrigger** (exécuter le déclencheur) ci-dessus, mais expose des paramètres entrants supplémentaires qui acceptent les listes de paires formatées *nom-valeur*. Le déclencheur va analyser automatiquement la liste, extraire les valeurs et les sauvegarder dans les variables de même nom; donc pas besoin de créer de filtre d'extraction.
- **ExecuteTriggerAndSetVariablesWithResponse.** - Exécuter le déclencheur et définir les variables avec réponse. Similaire à **ExecuteTriggerWithResponse** (exécuter le déclencheur avec réponse) ci-dessus, mais expose des paramètres entrants supplémentaires qui acceptent les listes de paires formatées *nom-valeur*. Le déclencheur va analyser automatiquement la liste, extraire les valeurs et les sauvegarder dans les variables de même nom; donc pas besoin de créer de filtre d'extraction.

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

WSDL

Spécifie le style des messages SOAP. Il peut être soit un **Remote Procedure Call (RPC)** ou de

style **document**. Choisir le style compatible avec l'application qui procure les données à NiceLabel Automation.

Le document WSDL (Web Service Description Language) définit les paramètres d'entrées et sorties de Web Service.

Définir un déclencheur Web Service sur le port 12345, le déployer dans Automation Manager et ensuite le lancer. Son WSDL sera disponible en :

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

Le WSDL présente les différentes méthodes qui fournissent des données au déclencheur. Il faut choisir le plus approprié.

- 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 des informations qui ne sont pas personnalisables. Le rapport contiendra des messages d'erreur par défaut.
- Les méthodes qui ont *SetVariables* (définir variables) dans leurs noms permettent de fournir une liste de variables en deux formats prédéfinis. Leurs valeurs seront extraites et mappées automatiquement aux variables appropriées. C'est du temps gagné 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.

L'interface Web Service définit les méthode 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) :

- **text.** (texte) C'est la chaîne de caractères d'entrée, qui peut être analysée par le filtre défini dans la configuration. Généralement la chaîne de caractères est structurée en CSV ou XML pour permettre une analyse facile par un filtre, mais n'importe quel format de texte peut être utilisé.
- **wait.** (attendre) C'est un champ Booléen qui spécifie s'il faut attendre la réponse de l'état d'impression et si le Service Web doit renvoyer des informations. Pour *Vrai* utiliser `1`, pour *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 (Il faut demander leurs valeurs en plaçant **Wait to** `1`) :

- **ExecuteTriggerResult** - Exécuter le résultat déclencheur. La réponse en nombre entier contiendra la valeur 0 s'il n'y a pas eu de problèmes de traitement des données, et elle contiendra un nombre entier supérieur à 0, quand une/des erreur(s) est/sont survenue(s). L'application qui exécute l'appel du Web Service NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **errorText.** - Texte d'erreur Cette chaîne de caractères contiendra les informations sur le travail d'impression, si une erreur est survenue durant le 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 quand 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_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>
```

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

- **text.** (texte) C'est la chaîne de caractères d'entrée, qui peut être analysée par le filtre défini dans la configuration. Généralement la chaîne de caractères est structurée en CSV ou XML pour permettre une analyse facile par un filtre, mais n'importe quel format de texte peut être utilisé.
- **wait.** (attendre) C'est un champ Booléen qui spécifie s'il faut attendre la réponse de l'état d'impression et si le Service Web doit renvoyer des informations. Pour *Vrai* utiliser `1`, pour *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 (Il faut demander leurs valeurs en plaçant **Wait to** `1`) :

- **ExecuteTriggerWithResponseResult** - Exécuter le déclencheur avec réponse résultat. La réponse en nombre entier contiendra la valeur 0 s'il n'y a pas eu de problèmes de traitement des données, et elle contiendra un nombre entier supérieur à 0, quand une/des erreur(s) est/sont survenue(s). L'application qui exécute l'appel Web Service à NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **responseData.** - Donnée de réponse La réponse personnalisée qui peut être définie dans la configuration du déclencheur Web Service. La réponse est encodée en base 64.
- **errorText.** - Texte d'erreur Cette chaîne de caractères contiendra les informations sur le travail d'impression, si une erreur est survenue durant le 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 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>
```

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

- **text.** (texte) C'est la chaîne de caractères d'entrée, qui peut être analysée par le filtre défini dans la configuration. Généralement la chaîne de caractères est structurée en CSV ou XML pour permettre une analyse facile par un filtre, mais n'importe quel format de texte peut être utilisé.
- **wait.** (attendre) C'est un champ Booléen qui spécifie s'il faut attendre la réponse de l'état d'impression et si le Service Web doit renvoyer des informations. Pour *Vrai* utiliser `1`, pour *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 va lire toutes les paires et assigner les valeurs fournies aux variables ayant le même nom. Si la variable n'existe pas dans le déclencheur, la paire *nom:valeur* est éliminée. Avec cette méthode pour fournir la liste de variables et leurs valeurs, il ne faut pas définir d'extraction de données avec les filtres. Le déclencheur se chargera de faire l'analyse.

  Le contenu de variableData peut être fourni dans une des deux structures disponibles.

  Structure XML

Les variables sont fournies dans l'élément racine `<Variables />` du fichier XML. Le nom de variable est fourni avec le nom d'attribut, la valeur de variable est fournie par l'élément valeur.

```
<?xml version="1.0" encoding="utf-8"?>
<Variables>
<variable name="Variable 1">Valeur 1</variable>
<variable name="Variable 2">Valeur 2</variable>
<variable name="Variable 3">Valeur 3</variable>
</Variables>
```

NOTE: Il faudra mettre 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 de caractères, pas comme une balise. Tout le contenu est utilisé comme données de caractères, par exemple `<element>ABC</element>` sera interprété comme `&lt;element&gt;ABC&lt;/element&gt;`. Une section CDATA commence par la séquence `<![CDATA[` et se termine par la séquence `]]>`. Placer simplement les données XML entre ces séquences.

Structure délimitée

Les variables sont fournies dans un flux de données. Chaque paire *nom:valeur* est fournie sur une ligne séparée. Le nom de variable se situe à gauche du signe égal (=), la valeur de variable se situe à droite.

```
Variable 1="Valeur 1"
Variable 2="Valeur 2"
Variable 3="Valeur 3"
```

Il y a les variables de sortie optionnelles suivantes (vous recevez leurs valeurs, si vous les demandez en plaçant **attendre** à `1`) :

- **ExecuteTriggerAndSetVariablesResult** - exécuter le déclencheur et définir les variables résultat. La réponse en nombre entier contiendra la valeur 0 s'il n'y a pas eu de problèmes de traitement des données, et elle contiendra un nombre entier supérieur à 0, quand une/des erreur(s) est/sont survenue(s). L'application qui exécute l'appel du Web Service NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **errorText.** - texte d'erreur Cette chaîne de caractères contiendra les informations sur le travail d'impression, si une erreur est survenue durant le 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.

ExecuteTriggerAndSetVariablesWithResponse. - Exécuter le déclencheur et définir les variables avec réponse.

Cette méthode sera utilisée quand 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) :

- **text.** (texte) C'est la chaîne de caractères d'entrée, qui peut être analysée par le filtre défini dans la configuration. Généralement la chaîne de caractères est structurée en CSV ou XML pour permettre une analyse facile par un filtre, mais n'importe quel format de texte peut être utilisé.
- **wait.** (attendre) C'est un champ Booléen qui spécifie s'il faut attendre la réponse de l'état d'impression et si le Service Web doit renvoyer des informations. Pour *Vrai* utiliser `1`, pour *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 va lire toutes les paires et assigner les valeurs fournies aux variables ayant le même nom. Si la variable n'existe pas dans le déclencheur, la paire *nom:valeur* est éliminée. Avec cette méthode pour fournir la liste de variables et leurs valeurs, il ne faut pas définir d'extraction de données avec les filtres. Le déclencheur se chargera de faire l'analyse.

  Le contenu de variableData peut être fourni dans une des deux structures disponibles.

  Structure XML

  Les variables sont fournies dans l'élément racine `<Variables />` du fichier XML. Le nom de variable est fourni avec le nom d'attribut, la valeur de variable est fournie par l'élément valeur.

  ```
  <?xml version="1.0" encoding="utf-8"?>
  <Variables>
  <variable name="Variable 1">Valeur 1</variable>
  <variable name="Variable 2">Valeur 2</variable>
  ```

```
<variable name="Variable 3">Valeur 3</variable>
</Variables>
```

NOTE: Il faudra mettre 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 de caractères, pas comme une balise. Tout le contenu est utilisé comme données de caractères, par exemple `<element>ABC</element>` sera interprété comme `&lt;element&gt;ABC&lt;/element&gt;`. Une section CDATA commence par la séquence `<![CDATA[` et se termine par la séquence `]]>`. Placer simplement les données XML entre ces séquences.

Structure délimitée

Les variables sont fournies dans un flux de données. Chaque paire *nom:valeur* est fournie sur une ligne séparée. Le nom de variable se situe à gauche du signe égal (=), la valeur de variable se situe à droite.

```
Variable 1="Valeur 1"
Variable 2="Valeur 2"
Variable 3="Valeur 3"
```

Il y a les variables de sortie facultatives suivantes (Il faut demander leurs valeurs en plaçant **attendre** `1`) :

- **ExecuteTriggerAndSetVariablesWithResponseResult** - exécuter le déclencheur avec réponse résultat. La réponse en nombre entier contiendra la valeur 0 s'il n'y a pas eu de problèmes de traitement des données, et elle contiendra un nombre entier supérieur à 0, quand une/des erreur(s) est/sont survenue(s). L'application qui exécute l'appel du Web Service NiceLabel Automation peut utiliser la réponse comme indicateur d'erreur.
- **responseData.** - Donnée de réponse La réponse personnalisée qui peut être définie dans la configuration du déclencheur Web Service. La réponse est encodée en base 64.
- **errorText.** - texte d'erreur Cette chaîne de caractères contiendra les informations sur le travail d'impression, si une erreur est survenue durant le 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.

Autre

Les options de la section **Commentaires du moteur d'impression** précisent la communication avec le moteur d'impression.

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

- **Impression supervisée.** Active le mode d'impression synchronisée. 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, NiceLabel Automation va dire qu'il y a une erreur et interrompre le processus d'impression en cas d'enregistrement d'une valeur trop longue dans la variable d'étiquette, ou de paramétrage d'une valeur dans une variable inexistante.

- **Ignorer les contenus de variable excessifs.** Les valeurs dépassant la longueur de la variable définie dans l'éditeur d'étiquettes seront tronquées pour pouvoir entrer dans la variable. Cette option s'applique lors du paramétrage de valeurs 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 10 caractères. Quand cette option est activée, toute valeur de plus de 10 caractères sera tronquée au 10 premiers caractères, tous les caractères suivant le caractère numéro 10 seront ignorés.

- **Ignorer les variables d'étiquettes manquantes.** En imprimant avec des fichiers de commande (tel qu'un fichier JOB), le processus d'impression va ignorer toutes les variables spécifiées dans le fichier de commande (utilisant la commande SET), mais non définies dans l'étiquette. Il n'y aura pas d'erreur lors de l'essai de paramétrage de la variable d'étiquette inexistante. Un processus semblable s'effectue lorsqu'une zone d'affectation est définie dans le filtre pour extraire toutes les paires *nom:valeur*, mais qu'il y a moins de variables définies dans l'étiquette.

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

- **Langage de Script.** Spécifie le langage du script activé 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 dans un 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 dans un fichier.** Activer cette option pour enregistrer les données dans le déclencheur uniquement si l'erreur se produit durant l'exécution de l'action. Activer cette option pour récupérer les données qui ont causé l'erreur et résoudre le problème ultérieurement.

  ATTENTION : Il faut activer l'impression supervisée, sinon NiceLabel Automation ne sera pas en mesure de détecter l'erreur durant l'exécution. Pour plus d'informations, Consulter l'article Mode d'impression Synchrone.

NOTE: NiceLabel Automation enregistre toujours les données reçues dans un fichier temporaire, qui est effacé dès la fin d'exécution du déclencheur . La variable interne `DataFileName` pointe vers ce fichier. Pour plus d'informations, consulter l'article Variables Internes.

Sécurité

- **Verrouiller et encoder le déclencheur.** Active la protection du déclencheur. Quand elle est activée, le déclencheur est verrouillé et ne peut pas être édité; les actions deviennent encodées. Seul l'utilisateur ayant le mot de passe peut déverrouiller le déclencheur et le modifier.

6.3 Utilisation De Variables

6.3.1 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. Le filtre va extraire les valeurs des flux de données reçues par le déclencheur et les envoyer dans les variables. Pour plus de renseignements, consulter 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 pour envoyer des valeurs de variables aux étiquettes utilise le mappage de nom automatique - la valeur de variable définie dans le déclencheur sera envoyée à la variable de même nom définie dans l'étiquette. Il y a trois façons de définir les variables :

- **Importer les variables 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 garantit que les noms de variables correspondent et permet de gagner du temps. 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.
- **Définition manuelle des variables.** 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 seraient `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.

CONSEIL: 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).
- **Longueur limite de 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 faut limiter la longueur de variable pour certains objets sur l'étiquette, tel que le code à barres EAN-13, qui accepte 13 caractères.
- **Valeur requise.** Spécifie que la variable doit contenir une valeur
- **Valeur par défaut.** Spécifie une valeur par défaut. La valeur par défaut sera toujours utilisée si aucune valeur n'est assignée à la variable.

6.3.2 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 la valeur fixe destinée à la variable.

  ```
  C'est une valeur fixe.
  ```
- **Utiliser des valeurs fixes et des données de variables.** Définir les valeurs combinées, composées 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 pour utiliser des variables dans les actions.
Dans ce cas, le contenu sera formé de trois variables et de certaines données fixes.

```
[variable1] // C'est une valeur fixe [variable2][variable3]
```

- **Utiliser les caractères spéciaux.** Vous pouvez également ajouter des caractères spéciaux au mélange. Saisir les caractères spéciaux manuellement, ou les insérer. Pour plus d'informations, consulter l'article Introduire des caractères spéciaux (Codes de Contrôle).
  Dans ce cas, la valeur de la `variable1` fusionnera avec des données fixes et des caractères binaires de saut de page.

```
[variable1] Saut de page va suivre ce texte fixe <FF>
```

6.3.3 Variables Internes

Les variables internes sont pré-définies 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 des déclencheurs sont internes pour chaque déclencheur.

Variable Interne	Disponible dans le déclencheur	Description
ActionLastErrorDesc	Tous	Fournit la description de la dernière erreur survenue. Utiliser cette valeur dans un retour d'informations au système hôte pour identifier la cause du défaut.
ActionLastErrorID	Tous	Fournit l'ID de la dernière erreur survenue. C'est un nombre entier. Quand la valeur est 0, il n'a pas eu d'erreur. Utiliser cette valeur dans les conditions, pour évaluer si une erreur est survenue ou pas.
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 26.2.2013.
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 2013.
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 a répondu. 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 DataFileName, 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. Utile 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.

6.3.4 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. La variable globale va fournir 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 va seulement les utiliser. La source des variables globales est configurable dans **Options** (Fichier >Outils >Options).

Par défaut NiceLabel Automation est configuré pour utiliser les variables globales de l'ordinateur local. L'emplacement par défaut est le suivant:

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

Les variables globales sont définies dans les fichiers `GLOBAL.TDB` et `GLOBALS.TDB.SCH`.

Dans un environnement multi utilisateurs, il faut 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 **NiceLabel LMS Enterprise** et **NiceLabel LMS Pro**).

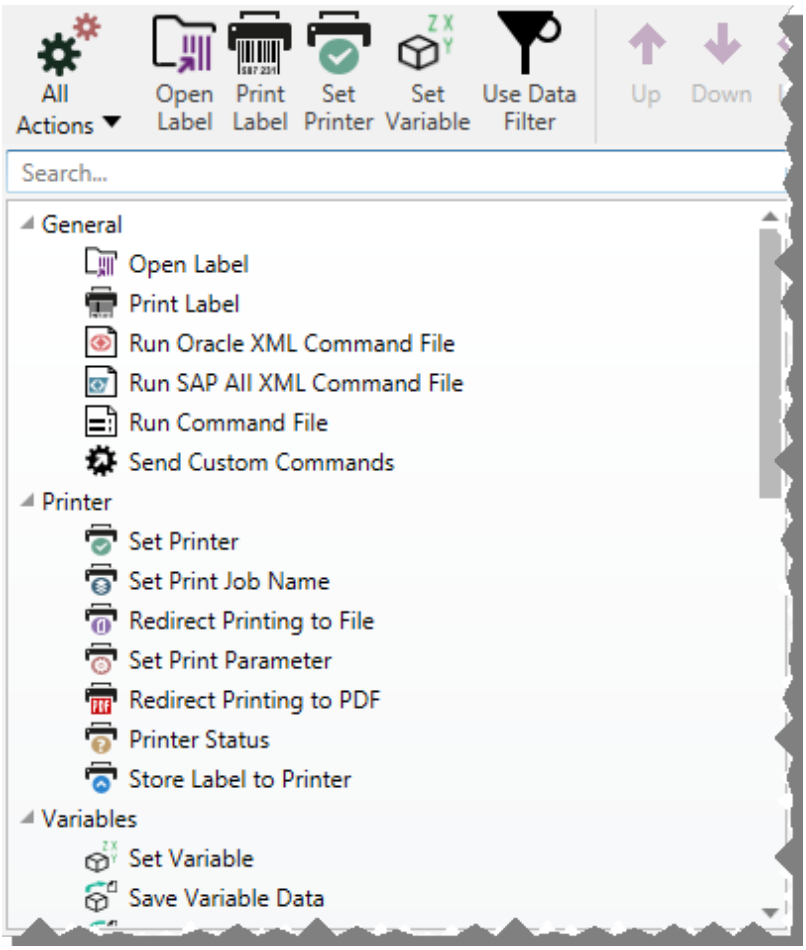
6.4 Utilisation Des Actions

6.4.1 Actions

La section Actions spécifie la liste d'actions qui seront exécutées chaque fois que le déclencheur est activé.

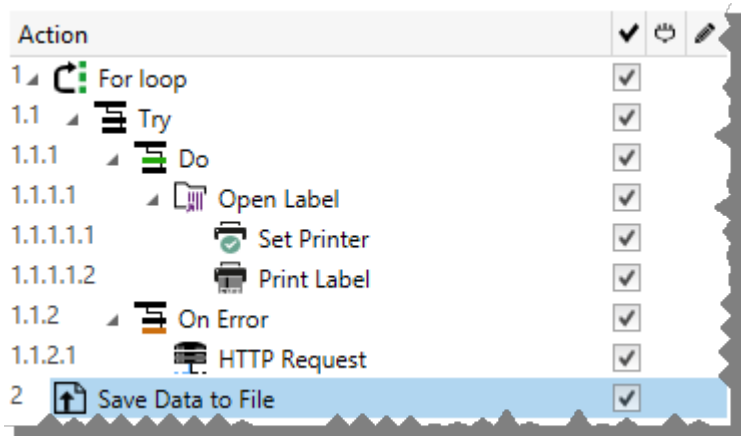
6.4.1.1 Définition Des Actions

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.



6.4.1.2 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.

6.4.1.3 Exécution D'Action.

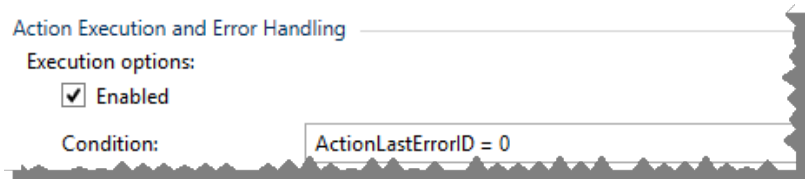
La liste des actions ne s'exécute qu'une fois par déclencheur. L'exécution des actions se fait de haut en bas, donc l'ordre des actions est important.

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 2017 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.

6.4.1.4 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 **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.

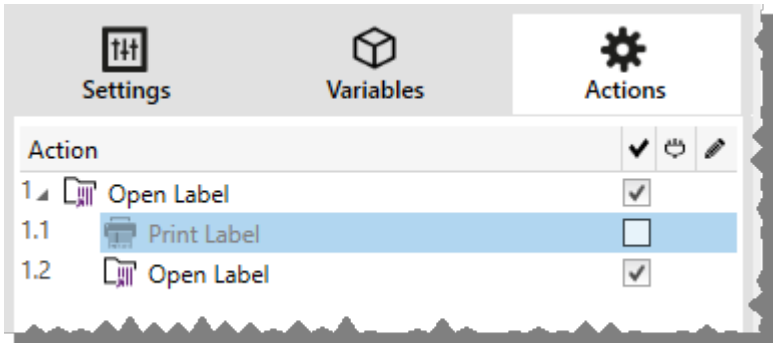
6.4.1.5 Identification Des Actions En État D'erreur De Configuration

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.

6.4.1.6 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.

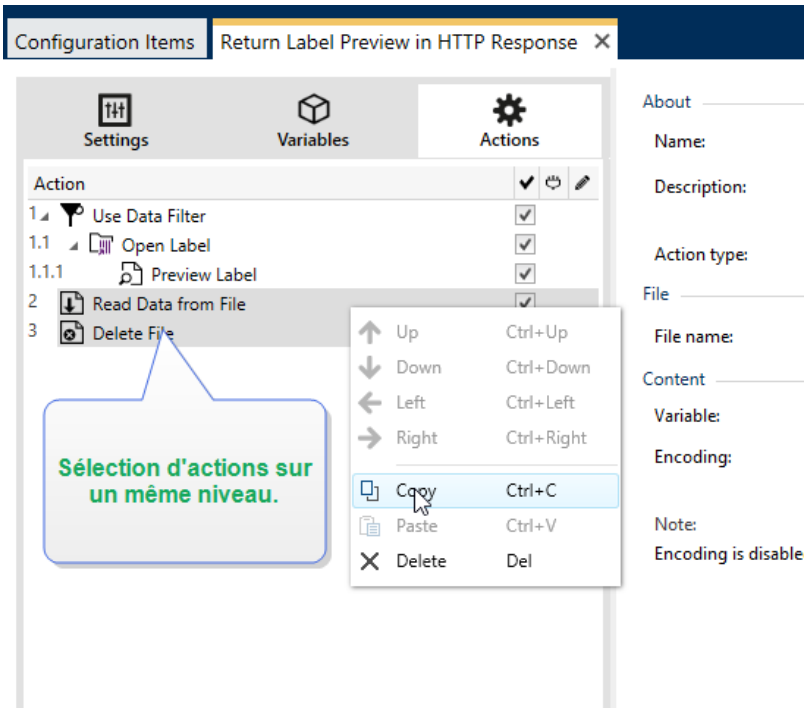
6.4.1.7 Copier Les Actions.

L'action peut être copiée et collée sur le même déclencheur ou sur un autre. Utiliser les raccourcis-clavier Windows standards 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.



6.4.1.8 Naviguer Dans La Liste D'actions.

Utiliser la souris pour sélectionner l'action définie et puis cliquer sur le bouton à 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 + Flêches vont déplacer la position de l'action vers le haut ou le bas, et à gauche ou à droite pour l'indentation.

6.4.1.9 Description Des Actions

Le groupe **A propos** permet de décrire toutes les actions NiceLabel 2017.

- **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.

6.4.2 Général

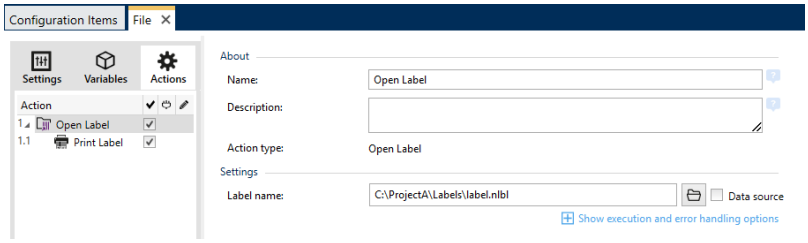
6.4.2.1 Open Label - Ouvrir L'étiquette

L'action **Ouvrir l'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, ensuite il ouvrira l'étiquette.

Dans cet exemple, NiceLabel 2017 charge l'étiquette `label.nlbl` du dossier `C:\ProjectA\Labels`.

```
C:\ProjectA\Labels\label.nlbl
```



S'il ne trouve pas l'étiquette spécifiée, NiceLabel 2017 essaye de la trouver à un autre endroit. Pour plus d'informations, se référer à :

A 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 2017 permet d'utiliser des chemins relatifs pour référencer les fichiers d'étiquettes. Le dossier de configuration (MISX) est toujours stocké dans le dossier racine.

En utilisant la syntaxe suivante, l'étiquette se chargera de façon relative depuis l'emplacement du fichier de configuration. L'étiquette sera recherchée dans le dossier `ProjectA`, ce qui est deux niveaux plus haut que le dossier actuel, 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.

  CONSEIL: 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.2.2 Impression De 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 2017 active deux paramètres par défaut :

- **Processus 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 le chapitre 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'infos à l'application fournissant les données. Pour plus d'informations, consulter le chapitre 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 la section Impression en session dans la guide utilisateur de NiceLabel Automation.

A 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.

- **Etiquettes:** 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. Pour dire à l'imprimante d'imprimer la même étiquette continuellement jusqu'à ce qu'elle soit coupée, ou qu'elle reçoive une commande d'effacement de sa mémoire tampon.

   ATTENTION : Ce scénario a besoin qu'un pilote d'imprimante NiceLabel soit installé et utilisé.

   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 sert de signal que "l'événement soit 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 les données. 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 seule 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 d'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 nombres 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.

CONSEIL: 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.2.3 Exécuter Le Fichier De Commande Oracle XML

INFO NIVEAU DE PRODUIT DESIGNER : La fonctionnalité décrite se trouve dans le module Automation Builder de **NiceLabel LMS Enterprise** et **NiceLabel LMS Pro.**

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.

Pour pouvoir utiliser cette action, le fichier XML doit être conforme au spécifications Oracle XML. Pour plus d'informations, consulter le chapitre Spécifications d'Oracle XML dans le guide utilisateur de NiceLabel Automation.

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.

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. In **Actions** tab, add the action to execute the command file, such as Exécuter le fichier de commande Oracle XML, Run Oracle XML Command File, or Run SAP All XML Command File (last two actions are available in Automation Builder).

   Pour l'action xécuter un **Fichier de Commande**, sélectionnez 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.

A 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.2.4 Exécuter Le Fichier De Commande SAP All XML

INFO NIVEAU DE PRODUIT DESIGNER : La fonctionnalité décrite se trouve dans le module Automation Builder de **NiceLabel LMS Enterprise** et **NiceLabel LMS Pro.**

Cette action exécute l'impression avec les données d'un fichier au format SAP All XML.

NiceLabel Automation est directement compatible avec les fichiers XML ayant la structure " SAP All XML", qui sont définis par le logiciel SAP.

Cette action est un raccourci. Elle permet d'exécuter directement des fichiers SAP All XML sans avoir besoin de les analyser avec un filtre. Pour pouvoir utiliser cette action, le fichier XML doit être conforme aux spécifications SAP All XML. Pour plus d'informations, consulter le chapitre Spécifications SAP All XML dans le guide utilisateur de NiceLabel Automation.

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.

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. In **Actions** tab, add the action to execute the command file, such as Exécuter le fichier de commande SAP All XML, Run Oracle XML Command File, or Run SAP All XML Command File (last two actions are available in Automation Builder).

   Pour l'action xécuter un **Fichier de Commande**, sélectionnez 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.

A 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 All XML à utiliser

- **Nom du fichier** sélectionne le fichier de commande SAP All 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.2.5 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 2017 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 fchiers de commande.

A 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 de 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 l'article Accès aux Ressources de Réseau Partagées dans le guide utilisateur de NiceLabel Automation.

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. In **Actions** tab, add the action to execute the command file, such as Exécuter le fichier de commande, Run Oracle XML Command File, or Run SAP All XML Command File (last two actions are available in Automation Builder).

   Pour l'action xécuter un **Fichier de Commande**, sélectionnez 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.2.6 Envoyer Des Commandes Personnalisées

Cette action exécute les commandes personnalisées saisies dans NiceLabel.

Cette action doit toujours être indentée dans l'action Open Label - Ouvrir l'étiquette. Elle comporte toutes les références de l'étiquette à laquelle les commandes s'appliquent. Pour plus d'informations, consulter l'article Utiliser 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.

A 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.
- **Editeur 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.3 Imprimante

6.4.3.1 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 Open Label - 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. A défaut d'utiliser cette action **Installer l'imprimante**, l'étiquette s'imprime comme définie dans le masque d'étiquette.

ATTENTION : 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 types de caractères internes peuvent ne pas être disponibles pour l'imprimante sélectionnée.

A 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.3.2 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.

A 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.3.3 Rediriger L'impression Dans 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.

Cet 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 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.

Cette action **Rediriger l'impression vers un fichier** est également utile pour imprimer plusieurs étiquettes différentes (fichiers .LBL) sur l'imprimante réseau en conservant l'ordre d'étiquettes correct. Quand plusieurs fichiers .LBL sont imprimés par le même déclencheur, Automation Builder envoie chaque étiquette dans une 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'actionEnvoyer les données à l'imprimante.

A 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.

- **Ecraser 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.3.4 Définir Les Paramètres D'Impression

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.

Après l'action Définir l'imprimante pour changer l'imprimante, il faut ensuite utiliser l'action Configurer 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 n'est compatible qu'avec le DEwe

A 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.

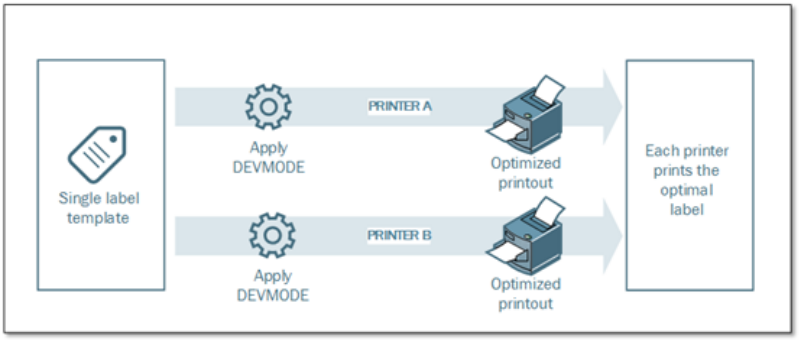
Le groupe **Paramètres d'impression** permet d'affiner les paramètres avant l'impression.

- **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 points 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:

CONSEIL: 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
- Rappelé 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 base 64 - 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 base 64 - 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**. 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.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.3.5 Rediriger L'impression Vers Un PDF

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.

Cet action demande au module NiceLabel 2017 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.

A 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.
- **Ecraser 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).

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.3.6 État De L'imprimante

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.

Scénarios possibles : (1) Vérification de l'état de l'imprimante avant l'impression. Si l'imprimante est en état d'erreur, imprimer l'étiquette sur l'imprimante en réserve. (2) Compter le nombre de travaux en attente dans le spouleur de l'imprimante principale. S'il y en a trop, imprimer sur une imprimante alternative. (3) Pour vérifier l'état de l'imprimante avant l'impression. Si l'imprimante est en état d'erreur, l'étiquette ne s'imprime pas, mais l'erreur est envoyée au système principal en utilisant une des actions de sortie, telles que Envoyer les données au port TCP/IP, Requête 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>pilotes>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 189.

A 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 **Etat de l'imprimante** .

ATTENTION : La majorité des paramètres suivants est compatible seulement avec les pilotes d'imprimante de NiceLabel. Avec un autre pilote d'imprimante, seuls les paramètres relatifs au spouleur sont utilisables.

- **Etat de l'imprimante:** spécifie l'état de l'imprimante en temps réel comme une chaîne de caractères.

  Si l'imprimante rapportent plusieurs états, ils sont fusionnés ensemble dans une chaîne de caractères, délimités par des virgules ",". Si aucun problème d'imprimante n'est reporté, le champs est vide. L'état de l'imprimante peut être: **Hors-ligne**, **Plus d'étiquettes** or **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 spooleur:** spécifie l'État du spooleur 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 spooleur:** spécifie l'état du spooleur sous forme de numéro rapportée 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.

  CONSEIL: 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 spooleur(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 spooler:** Spécifie le nombre de travaux qui sont dans le spouleur pour l'imprimante sélectionnée.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.3.7 Enregistrer L'étiquette Dans L'imprimante

Cette action enregistre un masque d'étiquette dans la mémoire d'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 de renseignements, consulter l'article Utiliser le mode d'impression Stocker/Rappeler dans le guide utilisateur de NiceLabel Automation.

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 elle est rappelée plus tard par un déclencheur logiciel ou mécanique, tel qu'un lecteur de codes à barres ou une cellule photoélectrique.

A 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:** Définit 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.

  ATTENTION : 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.4 Variables

6.4.4.1 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.

A 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.4.2 Enregistrer Les Données Variables

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.

CONSEIL: 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`.

A 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 l'article 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.

  CONSEIL: 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.4.3 Charger Les Données Variables

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.

A 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 l'article 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.

> **CONSEIL:** 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

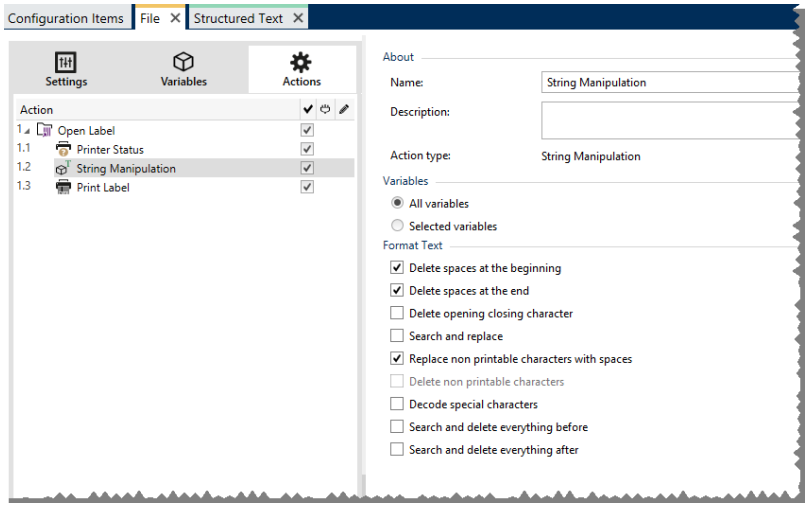
6.4.4.4 Manipulation De Chaîne De Caractères

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, effacer les parenthèses 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.



A 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 formatés.

- **Variables sélectionnées :**Spécifie que seules les variables sélectionnées dans le fichier de données seront formatées.

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 2017 utilise la syntaxe .NET Framework pour les expressions normales. Pour plus d'informations, consulter la Base de Connaissances article KB250.

- **Remplacer les caractères non imprimables par des espaces:** remplace tous les caractères de contrôle de la chaîne par un espace (decimal ASCII code 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:** efface 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: retour chariot ou passage à la ligne. NiceLabel 2017 utilise une notation pour encoder de tels caractères sous forme lisible, tels que <CR> pour Retour Charriot 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>", [[[Undefined variable Variables.Edition-Designer V7]]] 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` CR (Retour Chariot- code ASCII 13) et `LF` LF (Passage à la Ligne - code ASCII 10).

- **Rechercher et supprimer tout avant:** recherche la chaîne,e et supprime 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 supprime les caractères situés après la chaîne définie La chaîne de caractères trouvée peut aussi être effacée.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

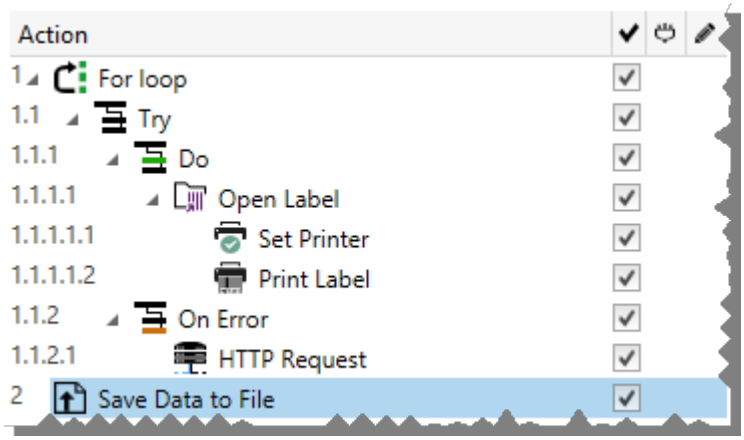
- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.5 Impression Par Lot

6.4.5.1 Boucler

INFO NIVEAU DE PRODUIT DESIGNER : 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 la section Impression en session dans la guide utilisateur de NiceLabel Automation.

A 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.

  CONSEIL: 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.5.2 Utiliser Un Filtre De Données

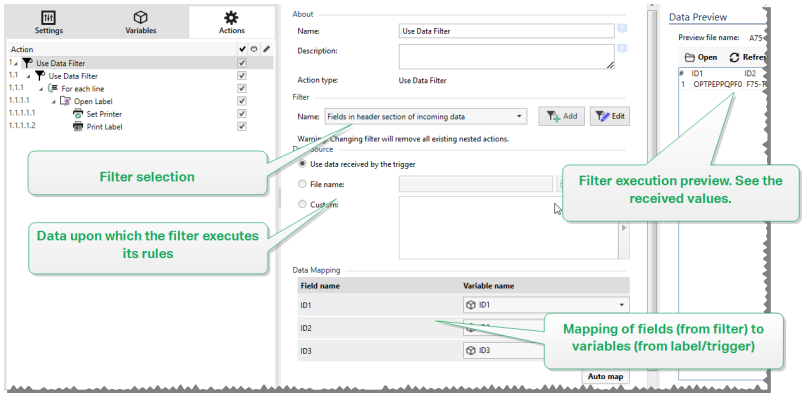
Cette action exécute les règles du filtre sur la source de données d'entrée. L'action va extraire les champs des données d'entrée et relier leurs valeurs aux variables associées.

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.
- **Relier les variables aux champs.** 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.

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 Automatique** va mapper les noms correspondants automatiquement.

- **Tester 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.



Pour plus de renseignements, consulter l'article Comprendre les Filtres et l'article 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 action 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 contplicenu 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 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. La section **Aperçu de Données** sera mise 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.5.3 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.

A 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.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6 Données Et Connectivité

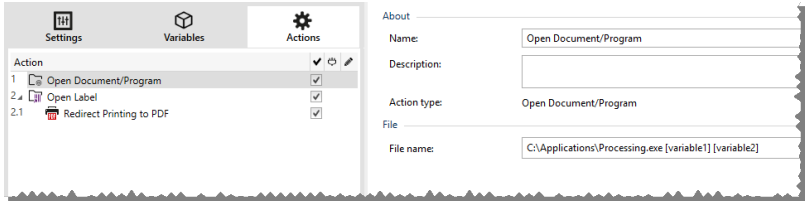
6.4.6.1 Ouvrir Un Document / Programme

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 le résultat à NiceLabel 2017. Cette action lui permet de se relier à un logiciel tiers qui peut traiter des données additionnelles, ou acquérir des données. Le logiciel externe peut fournir des réponses de données et les enregistrer dans un fichier, dans lequel elles seront récupérées pour des variables.

Fournir les valeurs de variable(s) au programme en les entourant de crochets dans la ligne de commande.

```
C:\Applications\Processing.exe [variable1] [variable2]
```



A 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 de 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.

- **Masquer la fenêtre:** rend invisible la fenêtre du programme ouvert. Comme NiceLabel 2017 fonctionne comme une application de service dans sa propre session, il 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 cette interaction dans Windows Vista et les systèmes d'exploitation plus récents pour raisons de sécurité.
- **Attendre la fin:** spécifie qu'il faut attendre la fin de cette action avant de continuer à exécuter les autres actions programmées en suivant.

  CONSEIL: 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.2 Enregistrer Les Données Dans Un Fichier

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 de Valeurs Composées 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. 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 flèche à droite de la zone de texte. Pour plus d'informations, consulter le chapitre 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.3 Lecture Des Données D'un Fichier

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.

ATTENTION : 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 de 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 qu'elles.

Le groupe **Ressayer après échec** définit comment l'action peut continuer si le fichier spécifié devient inaccessible

CONSEIL: 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, ou autre chose a échoué.

- **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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.4 Effacer Un Fichier

INFO NIVEAU DE PRODUIT DESIGNER : 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.

A 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 à 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.

  L'option **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 de 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.5 Exécuter Une Requête SQL

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.

  Exemples de Code SQL :

  ```
  SELECT Price FROM Products
    (sélectionner Prix de Produits) 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.

  Exemples de Code SQL :

  Changer le champ `AlreadyPrinted` (déjà imprimé) en `True` (vrai) pour l'enregistrement en cours de traitement.

  ```
  UPDATE Products
    (mise à jour produits) 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
 (effacer de Produits) 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.

A 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.

CONSEIL: Avant d'envoyer une requête SQL à une base de données, il faut paramétrer la connexion à la base de données. Cliquer 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.

CONSEIL: 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. **Aperçu des données** Cette section permet de tester l'exécution des requêtes SQL avec des données réelles.

  CONSEIL: 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, ensuite la transactions sera 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.
- **Monde Exécution:** spécifie le mode explicite d'exécution de la requête SQL.

CONSEIL: Avec certaines requêtes SQL complexes, il devient très difficile de déterminer automatiquement quelle est l'action prévue. Si la logique intégrée pose problème pour identifier l'objet de l'action, sélectionner l'action principale à la main.

  - **Automatique:** détermine l'action automatiquement.
  - **Retourner un ensemble d'enregistrement (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 enregistrement, supprimer ou mettre à jour les enregistrements existants. Le résultat est une réponse sur le nombre de lignes affectées par votre requête.

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 l'instruction 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 ligne suivantes contiennent les enregistrements.

    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 instructions 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, une nouvelle action Pour chaque enregistrement, s'ajoute automatiquement. Toutes les actions indentées sont répétées pour chaque enregistrement renvoyé par la requête SQL.

  NOTE: Le mappage automatique est activé. L'action Pour chaque enregistrement ne peut pas être supprimée.

Réessayer après échec .Ce groupe permet de configurer l'action de 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 d'essai de connexion au serveur de la base de données.
- **Intervalle entre les tentatives :** spécifie le temps d'attente entre chaque essai.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.6 Envoyer Les Données Au Port TCP/IP

INFO NIVEAU DE PRODUIT DESIGNER : La fonctionnalité décrite se trouve dans **NiceLabel LMS Enterprise** et **NiceLabel LMS Pro.**

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.

A 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.

- **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 le chapitre 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, entrer la destination suivante :
  `[hostname]:[port] (nom d'hôte)`

- **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 la flèche 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 reçu 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.

CONSEIL: 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 le chapitre 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.7 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.

CONSEIL: 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.

A 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.

CONSEIL: 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 le chapitre 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.8 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.

A 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 se 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 la flèche 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 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 :** Point de départ des données extraites.
- **Position finale :** point final des données extraites.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.9 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.

Scénario possible : Les données reçues par le déclencheur doivent être imprimées sur la même imprimante réseau mais sur des masques d'étiquettes différents (.NBLN Files). 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 dans un fichier pour envoyer ensuite un seul gros travail d'impression à l'imprimante.

A 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 flèche à droite de la zone de données et insérer la variable de la liste. Pour plus d'informations, consulter le chapitre Combinaison de valeurs dans un objet, dans le guide utilisateur de NiceLabel 2017.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.10 Requête HTTP

Cette action envoie les données au serveur Web de destination en utilisant la méthode HTTP sélectionnée. Les shémas d'URI HTTP et HTTPS sont autorisés.

HTTP fonctionne comme un protocole de requête-réponse entre client et serveur. Par cette action, NiceLabel 2017 prend le rôle de client, communiquant avec le serveur distant. 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.

A 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.

CONSEIL: 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 de 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, vous pouvez entrer la destination suivante :
`[hostname]:[port] (nom d'hôte)`

- **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.

CONSEIL: 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.

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 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 flèche à droite de la zone de données et insérer la variable de la liste. Pour plus d'informations, consulter le chapitre Combinaison de valeurs dans un objet, dans le guide utilisateur de NiceLabel 2017.
- **Encodage:** Spécifie l'encodage des données envoyées.

CONSEIL: 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 le chapitre Combinaison de valeurs dans un objet, dans le guide utilisateur de NiceLabel 2017.

  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.

  Utiliser la syntaxe suivante pour saisir les entêtes HTTP :

  ```
  Nom du champ d'entête : valeur du champ d'entête
  ```

  Par exemple, pour utiliser les noms du champ d'entête `Accept`, `User-Agent` and `Content-Type`, il faut utiliser la syntaxe suivante :

  ```
  Accept: application/json; charset=utf-8
  User-Agent: 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
  ```

  Les noms des champs d'entête peuvent être codés en dur ou bien ils peuvent récupérer leur valeur des variables du déclencheur. 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.6.11 Service Web

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.

Il faut imprimer des étiquettes de produit. Le déclencheur ne doit recevoir que le segment de données nécessaire. Par exemple : 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.

A 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 2017 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.

  Elle 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 le chapitre Combinaison de valeurs dans un objet, dans le guide utilisateur de NiceLabel 2017.

- **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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7 Autre

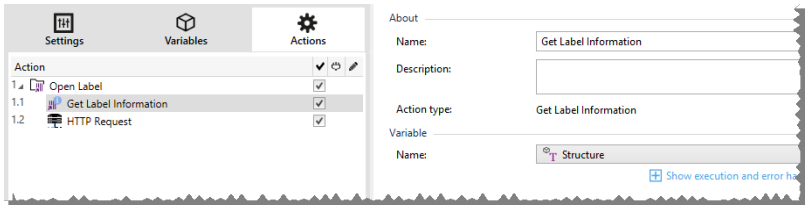
6.4.7.1 Récupérer Les Informations De L'étiquette

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'étiquettes renvoie les informations d'origine telles qu'enregistrées dans le fichier de l'étiquette. De plus elle fournir 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 c'est un déclencheur bidirectionnel.



NOTE: Cette action doit être indentée sous l'action Open Label - Ouvrir l'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.
    - Pour utiliser les informations de cet XML dans ce déclencheur, il faut définir le Filtre XML et l'exécuter avec l'action Utiliser le filtre de données.(seulement avec Automation Builder)
    - Pour renvoyer les données XML comme réponse du déclencheur HTTP ou Web Service, utiliser cette variable directement dans le champ **Données de réponse** dans la configuration du déclencheur.
    - Pour 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.

CONSEIL: Pour plus de renseignements sur les valeurs provisoires, consulter le guide utilisateur de NiceLabel 2017 Designer au chapitre Variables.

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>**: C'est la racine.
- **<Original>**:Spécifie les dimensions de l'étiquette et le nom d'imprimante tels qu'ils sont sauvegardés dans le fichier d'étiquette.
    - **Width** : Cet élément contient la largeur originale de l'étiquette.
    - **Height** : 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.

- **Width** : Cet élément contient la largeur actuelle de l'étiquette.
- **Height** : 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: L'imprimante utilisée sera différente si le pilote de l'imprimante originale n'est pas installé sur cet ordinateur, ou si l'imprimante est modifiée par l'action Définir l'imprimante.

- **<Variables> et <Variable>:** L'élément `Variables` contient la liste des variables saisies de l'étiquettes, chacune étant définie dans une élément `Variable` distinct. Les variables de saisie se trouvent listées dans la boite de dialogue Imprimer quand l'impression est effectuée depuis NiceLabel 2017. S'il n'y a pas de variable saisie dans l'étiquette, l'élément `Variables` est vide.
  - **Name:** Spécifie 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 de la création de l'étiquette.
  - **Format:** Contient le type de caractères possibles pour la variable.
  - **IsPrompted:** dit si la variable est saisie lors de l'impression ou non.
  - **PromptText:** contient le texte d'invite pour que l'utilisateur entre la valeur.
  - **CurrentValue:** Contient la valeur actuelle telle qu'elle sera utilisée pour l'impression.
  - **IncrementType:** Dit si la variable est définie comme compteur ou non. Si c'est un compteur, le type de compteur est précisé.
  - **IncrementStep:** contient l'information concernant le pas incrémental. La valeur du compteur s'incrémente/décrémente de cette valeur sur l'étiquette suivante.
  - **IncrementCount:** Donne le point de départ du compteur. Généralement, le compteur change de valeur à chaque étiquette, mais cela peut être modifié.
  - **Length:** détermine le nombre maximum de caractères stockés dans une variable.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

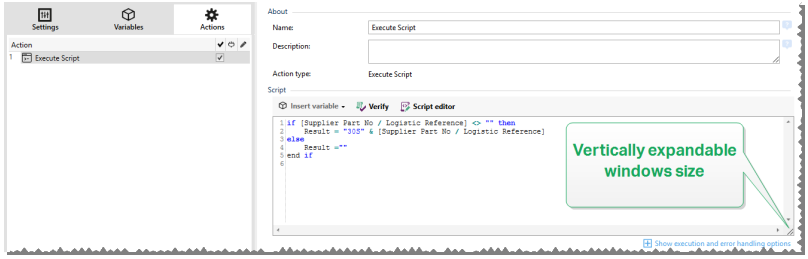
- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.2 Exécuter Un Script

L'utilisation de scripts VBScript ou Python personnalisés améliore les fonctionnalités du logiciel. Utiliser cette fonction si les actions pré définies 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. Pour plus d'informations, se référer à :



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.

A 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.
- **Editeur 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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.

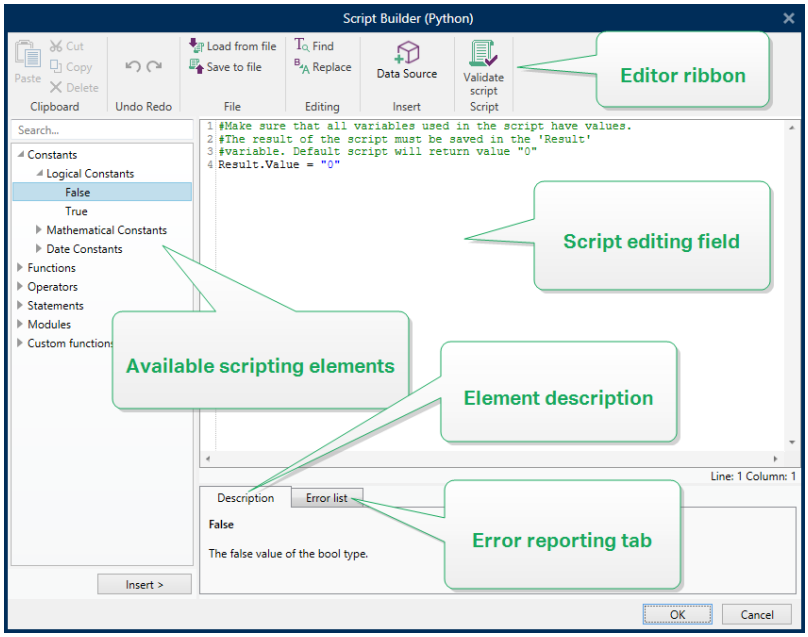
  NOTE: Les actions indentées qui dépendent de l'action actuelle ne seront pas exécutées. L'exécution des actions continuera avec l'action suivante qui se trouve au même niveau que l'action actuelle. L'erreur est enregistrée dans le journal, mais elle n'interrompt pas l'exécution de l'action.

EXEMPLE: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.2.1 Editeur De Script

NiceLabel 2017 comporte un éditeur de scripts qui facilite l'écriture en Python ou en VB, sans erreur ni perte de temps.



Le **ruban de l'éditeur** dispose des commandes usuelles qui sont réparties dans plusieurs groupes par fonction.

- Le groupe **Presse papier** 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 d'un fichier:** charge un script d'un ficher texte externe enregistré auparavant.
  - **Enregistrer dans un fichier:** stocke le script en cours dans un fichier texte.
- Le groupe **Edition**: permet de rechercher et remplacer des chaînes dans un script.
  - **Rechercher:** localise la chaîne saisie dans le script.
  - **Remplacer:** Remplace ce 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.

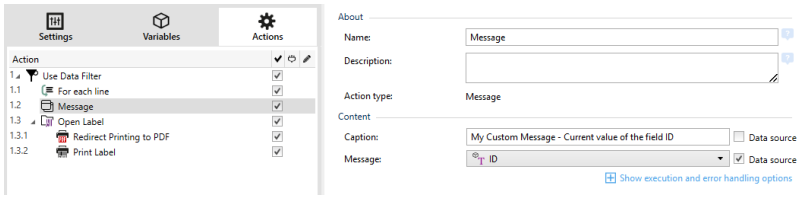
La **liste d'erreurs** comporte les erreurs rapportées après utilisation de la commande **Valider le script**

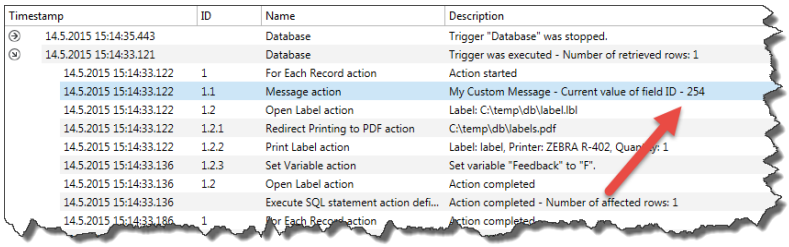
6.4.7.3 Message (configuration)

Cette action écrit un enregistrement personnalisé dans le journal.

Généralement, le fichier contient des chaînes de caractères et descriptions d'erreurs générées par l'application. Utiliser cette action pour écrire des chaînes de caractères personnalisées. C'est utile pour la résolution de problèmes et erreurs de configuration, cela permet de surveiller les valeurs des variables sélectionnées.

EXEMPLE: Consulter la copie d'écran ci-dessous pour configurer l'inscription dans le journal d'un message personnel dans le panneau du journal de Automation Builder (lors du test de la configuration) ou dans le panneau du journal de Automation Manager (après démarrage du déclencheur).





Le groupe **Contenu** définit l'intitulé et le contenu du message.

- **Intitulé :** spécifie le contenu du message personnel. L'option **Source de données** permet de définir le titre dynamiquement. Sélectionner ou créer une variable contenant le titre après exécution du déclencheur.
- **Message:** spécifie le contenu du message personnel. L'option **Source de données** permet de définir le contenu du message dynamiquement.

  CONSEIL: Un contenu dynamique est préparé à l'avance dans une autre action et utilisé ensuite ici.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.4 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.

CONSEIL: 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 2017.

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 autorisé puisent ouvrir la configuration. Pour plus d'informations, consulter le chapitre Protection de la configuration des déclencheurs, dans le guide utilisateur de NiceLabel Automation.

A 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 encodée dans la licence, les actions indentées ne s'exécutent pas.
    - Si la valeur introduite est 0, les actions s'exécuterons si elles trouvent une licence valide.

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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes

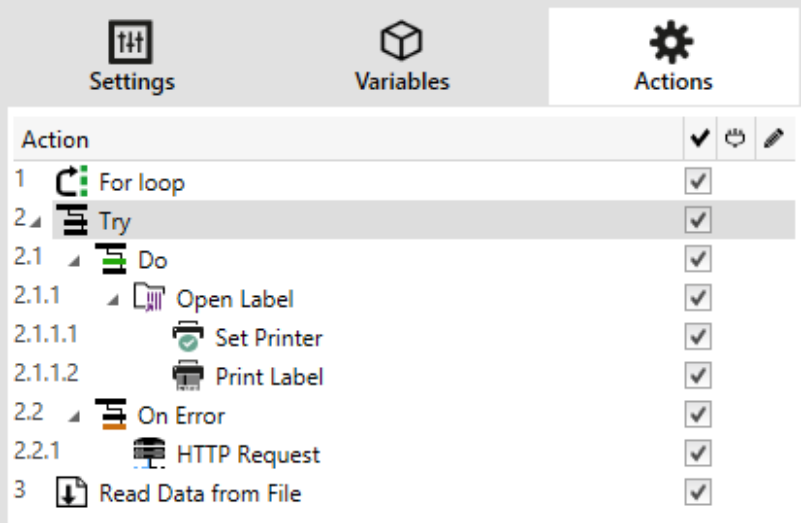
`ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.5 Essayer

Elle 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**.



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, 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 en 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.

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.

A 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.6 Transformation De L'XML

INFO NIVEAU DE PRODUIT DESIGNER : Cette fonctionnalité est disponible dans **NiceLabel 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.

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. Dans ce cas, 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. Il faut 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.

CONSEIL: 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** .

A 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 group e**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 (.XLST).
- **Personnalisé :** Définit le contenu personnalisé. Le contenu 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.7 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.

A 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.8 Consigner Les Événements

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.

A 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. Cet espace peut contenir jusqu'à 2000 caractères.

CONSEIL: La description saisie dans les champs **Information** et **Détails** permet de filtrer les événements dans **l'historique de toutes les activités** du 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.9 Aperçu De L'étiquette

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.

A 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. L'option **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.

CONSEIL: Valeur provisioire 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.4.7.10 Créer Une Variante D'étiquette

Elle 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 approuve 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 utilisées pour l'impression.

CONSEIL: 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 du NiceLabel 2017 Control Center.

A 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 est fournie au moment de l'impression. Exemples: les sources de données de production comme le lot, la date d'expiration, etc.

CONSEIL: Mettre seulement 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 document de 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 2017 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 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 document 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 :

- **Activé.** 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 de programmation en-ligne qui doit donner une valeur Booléenne (`vrai` ou `faux`). Quand le résultat de l'expression est `vrai`, l'action s'exécute. La condition permet d'éviter l'exécution des actions à chaque fois.

Options de **Traitement d'erreurs** :

- **Ignorer l'échec** : Pour préciser 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: A la fin de l'impression il est possible d'envoyer la mise à jour du statut 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 I**mprimer l'Étiquette** doit avoir l'option **Ignorer l'échec** activée.

- **Save error to variable:** enables the user to define the **Data Source** (variable) to save the error to. La même cause d'erreur est aussi enregistrée dans les variables internes `ActionLastErrorId` et `ActionLastErrorDesc`.

6.5 Test Des Déclencheurs

6.5.1 Test Des Déclencheurs

Quand la configuration du déclencheur est terminée, la moitié du travail est effectuée. Avant de déployer le déclencheur, il faut convenablement tester les opérations qu'il doit effectuer en fonction des données entrées et vérifier l'exécution des actions.

Le test de la configuration peut être effectué tout au long du développement de Automation Builder. 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 utilisant des vrais déclencheurs, tout en surveillant l'exécution du déclencheur dans Automation Manager.

Test d'exécution des actions individuelles

Certaines actions ont une fonctionnalité de prévisualisation qui permet de changer les paramètres d'entrée et de voir le résultat de l'action à l'écran.

- **Utiliser un Filtre de Données.** L'action montrera un aperçu réel des données analysées. Les règles du filtre sélectionné s'appliquent au fichier de données d'entrée sélectionné et 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 affichera un aperçu de l'exécution de la requête SQL définie, avec 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 et toutes les modifications peuvent être annulées. Les paramètres de requête peuvent être modifiés pour voir comment ils influencent le résultat.

- **Web Service.** L'action affichera un aperçu d'exécution de la méthode sélectionnée (fonction) du Web Service. Les paramètres de requête peuvent être modifiés pour voir comment ils influencent le résultat.
- **Exécuter le 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, quelque soit son type. Le déclencheur ne s'active pas au changement de l'événement surveillé, seul le déclencheur activé dans le Automation Manager peut le faire. En fait, le déclencheur effectue les actions en fonction des données enregistrées dans un fichier. Vérifier l'existence du fichier contenant les données du déclencheur 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, y compris le nombre et le contenu des étiquettes. Cela donnera 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 que ,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

C'est une bonne pratique de déployer la configuration dans Automation Manager sur un serveur de pré-production, avant le déploiement sur le serveur de production. Tester dans un environnement de pré-production permet d'identifier des problèmes de configuration qui n'ont pas été détectés durant les essais du déclencheur sur Automation Builder seul. La performance peut aussi être mise à l'épreuve en ajoutant la charge sur le déclencheur pour voir le résultat. Les tests fourniront d'importantes informations concernant la bande passante disponible et identifieront les points faibles. En fonction des conclusions, il est possible ensuite

d'implémenter diverses techniques d'optimisation, telles que optimiser la conception d'étiquette 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 et l'aperçu dans Automation Builder

Il ne faut pas se fier uniquement à l'aperçu du déclencheur à l'écran dans l'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 plate-forme 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 quand le traitement du déclencheur est affecté par les différences de plate-formes (32 bits vs 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 base de données. Pour plus d'informations, consulter l'article 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 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.

6.6 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.

6.7 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.

ATTENTION : 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.

6.8 Utilisation De La Couche De Transport Sécurisée (HTTPS)

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Protéger le trafic des données entrant sur le Déclencheur Serveur HTTP et sur le 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 générer 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.

1. Ouvrir une fenêtre de Commande.
2. Taper **mmc** et appuyer sur la touche ENTER (il faut disposer des droits d'administrateurs).
3. Dans le menu Fichier, cliquer sur **Ajouter/ Supprimer un composant logiciel enfichable.**
4. Dans **Composants logiciels enfichables disponibles,** sélectionner **Certificats**.
5. Cliquer sur **Ajouter**.
6. Dans l'interface **Composant logiciel enfichable Certificats**, sélectionner **Un compte d'ordinateur** et cliquer sur **Suivant**.
7. Dans l'interface **Sélectionner un ordinateur**, cliquer sur **Terminer**
8. Dans l'interface **Ajouter ou supprimer des composants logiciels enfichables,** cliquer sur **OK**.
9. Dans la fenêtre racine de la console, développer **Certificats>Personnel**.
10. Cliquer à droite sur le dossier Certificats et sélectionner **Toutes les tâches>Importer**.
11. Suivre les instructions pour importer le certificat.

3. Extraire l'empreinte d'un certificat juste importé.
    1. En restant dans le MMC, double cliquer sur le certificat.
    2. Dans l'interface du **Certificat**, cliquer sur l'onglet **Détails**.
    3. Dans la liste de champs, rechercher et cliquer sur Empreinte numérique.
    4. 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 va activer 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 le Automation Manager.

Désactiver la couche de transport sécurisée (HTTPS)

Dans le système Windows :

1. 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.

7 Exécuter et gérer les déclencheurs

7.1 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. A ce moment les déclencheurs prennent vie et commencent à surveiller les événements définis.

Pour déployer la configuration, utiliser les méthodes suivantes.

Déployer en utilisant 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 l'Automation Manager fonctionnant sur la même machine.
5. Démarrer les déclencheurs qu'il faut activer.

Si cette configuration était déjà chargée, le déploiement forcera le rechargement, gardant l'état actif des déclencheurs.

Déployer en utilisant Automation Manager

1. Démarrer Automation Manager.
2. Aller sur 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, voir l'article Contrôler le Service avec les paramètres de ligne de commande.

7.2 Options De Journalisation Des Événements

ATTENTION : Certaines fonctionnalités de ce chapitre 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 dans tous les produits NiceLabel Automation.

- **Journalisation dans la base de données.** Le journal dans une base de données interne est toujours actif, il enregistre tous les évènements avec tous les détails. Pour visualiser les informations journalisés, utiliser un filtre pour afficher les évènements correspondants. 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 évènements sont supprimés de la base de données une fois par semaine. Cette période est configurable dans les Options. Les données des anciens évènements sont effacées de la base de données, mais la base de données n'est pas compactée (vidée), elles peuvent donc encore occuper de la place sur le disque. Pour la compacter, utiliser un logiciel tiers de gestion SQLite.
- **Journalisation dans le journal d'événements de Windows.** Les évènements important sont sauvegardés dans le journal de Windows dans le cas où le NiceLabel Automation n'a pas pu démarrer, ce qui procure un journal secondaire.
- **Journal du Control Center.** Le journal du Control Center se trouve dans les produits **NiceLabel LMS Enterprise** et **NiceLabel LMS Pro**. Control Center est une console de gestion Web qui enregistre tous les événements des serveurs NiceLabel Automation. Les données sont sauvegardées dans la base de données du Serveur SQL de Microsoft. Control Center dispose aussi de recherche dans les données collectées, d'alertes automatisées pour certains évènements, d'une gestion des imprimantes, d'un stockage de documents, d'un système de contrôle de versions, de flux de travail et de la réimpression d'étiquettes.

  NOTE: Pour plus d'informations, consulter le Guide d'utilisation du Control Center.

7.3 Gestion Des Déclencheurs

L'application Automation Manager est la partie de gestion du logiciel NiceLabel Automation. Utiliser Automation Builder pour configurer les déclencheurs, et 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.

En activant l'affichage **par état**, les déclencheurs de toutes les configurations ouvertes qui correspondent à l'état choisis s'affichent ensemble. En activant l'affichage **par configuration**, les déclencheurs de la configurations 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 vont changer 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, l'Automation Manager ne l'applique pas automatiquement. Pour appliquer le changement, il faut 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 seront chargés en état suspendu. Il faut alors démarrer les déclencheurs pour les activer. Pour plus d'informations, voir l'article 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

Quand la configuration est mise à jour et enregistrée dans Automation Builder, les changements ne s'appliquent pas automatiquement dans Automation Manager. Pour recharger la configuration, cliquer à droite sur le nom de configuration, ensuite sélectionner **Recharger la Configuration**. Tous les déclencheurs seront rechargés. Si la mise en cache des fichiers est activée, le rechargement va forcer la synchronisation de tous les fichiers utilisés par les déclencheurs.

Démarrer / arrêter les déclencheurs

Lors du chargement des déclencheurs 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 le bouton **Arrêter**. Plusieurs déclencheurs d'une même configuration peuvent être démarrer / arrêter simultanément.

Une ligne de commande peut aussi contrôler le démarrage/arrêt. Pour plus d'informations, voir l'article Contrôler le Service avec les paramètres de ligne de commande.

Gestion des conflits de déclencheurs

Les déclencheurs peuvent être en erreur à cause des situations suivantes. Un déclencheur en erreur ne peut pas démarrer avant que le problème soit résolu .

- **Erreur de configuration du déclencheur ou configuration incomplète.** 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 des données sur le même port. Si la configuration du déclencheur se croise avec un autre déclencheur, le second déclencheur ne fonctionnera 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, l'icône du déclencheur change en couleur rouge, le déclencheur est en état d'erreur et les détails de l'événement sont sauvegardés dans le journal de la base de données. Même si les événements suivants se terminent avec succès, le déclencheur reste en état d'erreur jusqu'à confirmation de 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 dans 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,** tel 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.

7.4 Utilisation Du Journal D'événements

Toutes les activités du logiciel NiceLabel Automation sont enregistrées dans la base de données du journal pour historique et dépannage. En cliquant sur le bouton **Journal** sur l'onglet des déclencheurs, les événements de ce déclencheur s'affichent. Le journal affiche les informations concernant tous les événements correspondant au filtre défini.

Les données du journal sont utiles pour la résolution de problèmes. Si l'action du déclencheur ne peut pas être exécutée, l'application enregistre une description de l'erreur dans le fichier du journal, ce qui permettra d'identifier et résoudre le problème.

NOTE: Par défaut, la durée de détention des données est de 7 jours. Elle peut être modifiées 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 évènements afficher, les évènements du déclencheur sélectionné ou les évènements de tous les déclencheurs de la configuration sélectionnée.
- **Période enregistrée dans le journal.** Spécifie la période de temps durant laquelle les évènements se sont déroulés. Par défaut**, Les 5 dernières minutes**.
- **Niveau de l'évènement.** Spécifie le type (l'importance) des évènements à afficher. **Erreur** est un type d'évènement qui va interrompre l'exécution. **Avertissement** est un type d'évènement dans lequel des erreurs surviennent mais elles sont configurées pour être ignorées. **Information** est un type d'évènement qui enregistre toutes les informations non-erronées. Le niveau du journal est configurable dans les Options.
- **Filtrer par texte.** Afficher tous les évènements qui contiennent une chaîne de caractères donnée. Utiliser cette option pour la résolution d'erreurs sur des déclencheurs chargés. Le filtre sera appliqué 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**.

ATTENTION : A utiliser avec précautions, car il n'y a pas de retour en arrière possible. Ceci enlèvera **TOUS** les évènements sauvegardés dans la base de données du journal, et s'appliquera à tous les déclencheurs, pas seulement le déclencheur en cours.

8 Performances et options de retour d'informations

8.1 Traitement Parallèle

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

La gamme de produits NiceLabel Automation a été développée pour permettre le traitement parallèle des processus entrants et sortants. Cela assure une efficacité maximale sur tous les systèmes sur lesquels le logiciel est installé. NiceLabel Automation peut effectuer plusieurs tâches simultanément, tout en conservant l'ordre dans lequel les déclencheurs sont arrivés. La rapidité des tâches d'impression est grandement dépendante du matériel sur lequel le logiciel est implémenté.

Traitement parallèle d'entrée

Plusieurs déclencheurs peuvent fonctionner sur la même machine, ils répondront tous simultanément aux changements dans les 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 aucune des tâches d'impression n'est actuellement disponible. Dès qu'une tâche d'impression est disponible, elle prend le premier flux d'impression qui se trouve dans la queue suivant le principe FIFO (Premier entré, Premier sorti). Cela permet de garantir le traitement des données entrées dans l'ordre correct . Mais ça ne garantit pas le principe FIFO pour l'impression. Voir paragraphe suivant.

NOTE: Non seulement plusieurs déclencheurs peuvent fonctionner en parallèle. 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, configurable par un masque.

Traitement parallèle de Sortie

Généralement le résultat du déclencheur est le processus d'impression de l'étiquette. Les données reçues sont utilisées et imprimées 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 appelées "cœurs". Les processeurs multiples peuvent exécuter des instructions multiples en même temps, ce qui augmente la vitesse totale de traitement, dans le cas du NiceLabel Automation, ils vont augmenter la vitesse de traitement des tâches d'impression, et donc améliorer les performances d'impression.

Par défaut, chaque produit NiceLabel Automation va exécuter les travaux d'impression dans un processus distinct sur chacun des cœurs qui est disponible dans la machine. Plus le processeur central est puissant, plus la capacité de traitement est élevée. Ceci optimise l'usage de la puissance disponible du processeur central. Durant l'installation, le logiciel installe des valeurs par défaut raisonnables pour que chaque cœur disponible traite une thread d'impression et normalement il n'y a pas besoin de les changer. Pour effectuer un changement, consulter l'article Changer les paramètres par défaut d'Impressions multi threads.

Avec une grande quantité de processus d'impression, les données du premier événement sont imprimées par un processus d'impression, alors 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. 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, consulter l'article Mode d'impression Synchrone.

8.2 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 consiste à conserver les fichiers déjà utilisés en mémoire. 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 activé par défaut. Le contenu de la mémoire cache sera vidé d'une configuration particulière, quand cette configuration est supprimée ou rechargée. 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 sauvegarde les données sur le disque. Il est destiné au stockage intermédiaire de fichiers. 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 les Options.

Prolongement des périodes pour le déchargement des étiquettes.

Une fois l'étiquette utilisée, elle est chargée dans la mémoire cache où elle est disponible pour impression instantanée la fois suivante. Le ménage de la mémoire cache enlève toutes les étiquettes non utilisées depuis 8 heures.

Pour prolonger ce temps dans la mémoire cache, procéder comme suit:

1. Rechercher le dossier système de NiceLabel Automation.

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

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 pendant lesquelles l'étiquette reste 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 doit avoir le contenu suivant :

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

6. Lors de l'enregistrement du fichier, le Service NiceLabel Automation va appliquer les paramètres automatiquement.

Activation du Cache persistant

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

Pour activer et configurer le cache persistant, ouvrir l'option, sélectionner NiceLabel Automation et activer **Mettre en cache les fichiers distants**

- **Actualiser les fichiers cache.** Définit l'intervalle de temps en minutes durant lequel le cache est synchronisé avec les fichiers du dossier d'origine. Ceci est l'intervalle de temps

durant lequel le système peut utiliser l'ancienne version du fichier.

- **Supprimer les fichiers du cache lorsqu'ils ont plus de.** Définit l'intervalle de temps en jours qui sera utilisé pour éliminer tous les fichiers du cache qui n'ont pas été utilisés durant cette période.

NiceLabel Automation utilise le dossier local suivant comme cache pour les fichiers distants :

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

NOTE: On peut mettre en cache des fichiers d'étiquettes ou d'images. Après avoir activer la mise en cache des fichiers, redémarrer le servie Automation pour que la modification prennent effet.

Forcer le rechargement du contenu du cache

NiceLabel Automation va mettre à jour automatiquement le contenu du cache à l'intervalle de temps défini (la valeur par défaut est de 5 minutes).

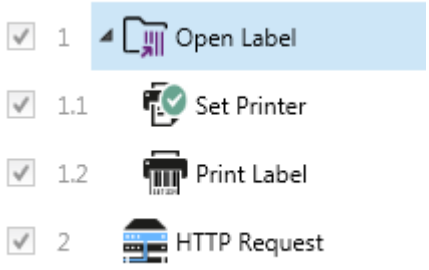
Pour forcer le rechargement du cache manuellement, effectuer les opérations suivantes:

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**.

8.3 Traitement D'Erreur

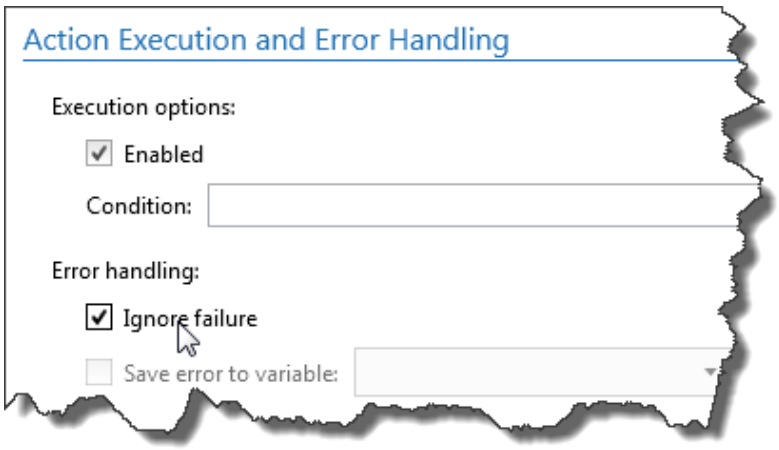
Quand une erreur survient durant l'exécution d'une action, NiceLabel Automation arrête l'exécution de toutes les actions du déclencheur. Les actions définies après l'action en cours ne seront pas exécutées.

Par exemple, les actions sont définies comme sur la copie d'écran. 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 seront 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'information doit être fait de façon asynchrone avec l'action **Requête HTTP** après la création du travail d'impression (ou pas). Après la fin du processus d'impression, il est possible de mettre à jour certaines applications en fonction du retour. Envoyer un message en format HTTP vers cette application.

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 de l'action Exécution et traitement des erreurs.



Si une action particulière échoue, NiceLabel Automation commencera l'exécution de l'action suivante dans le niveau précédent de la hiérarchie.

EXEMPLE: Par exemple, si l'action **Définir l'imprimante** dans le niveau 1.1 échoue, l'exécution ne continuera pas avec l'action **Imprimer l'étiquette** du niveau 1.2 car il échouera probablement aussi, mais va continuer par l'action **Requête HTTP** du niveau 2, car c'est l'action suivante dans le niveau supérieur de la hiérarchie.

La même logique peut être implémentée pour les actions répétitives, telles que **Utiliser un Filtre de Données**, **Boucler** et **Pour chaque enregistrement**, qui 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.

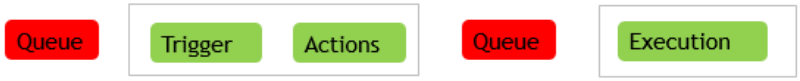
8.4 Mode D'impression Synchrone

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Mode D'impression Asynchrone

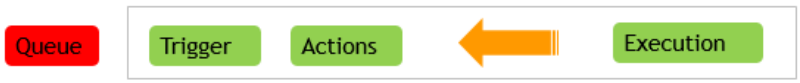
Par défaut, le mode d'opération de NiceLabel Automation est asynchrone. C'est une forme d'impression, dans laquelle un déclencheur envoie les données d'impression et ensuite ferme la connexion vers le sous-système d'impression. Le déclencheur n'attend pas les résultat du processus d'impression et ne 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 a une mémoire tampon devant lui, dans laquelle le déclencheur envoie les requêtes d'impression. La mémoire tampon absorbe les pointes et assure que les données ne sont pas perdues.

Si l'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 le mode d'impression asynchrone, il est impossible de définir l'exécution d'actions conditionnelles, si l'exécution du déclencheur est en erreur.



Mode D'impression Synchrone

Au contraire, le mode synchrone ne coupe pas la connexion avec le processus d'impression. Dans ce mode, le déclencheur envoie les données d'impression et reste connecté au sous-système d'impression tant qu'il traite les actions. Quand le processus d'impression se termine (avec succès ou avec une erreur), le déclencheur reçoit le retour d'information 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'information 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.

Il faut utiliser le mode synchrone pour recevoir le rapport d'information dans le déclencheur, ou pour assurer le mode d'impression FIFO (les données reçues dans les événements du déclencheur sont imprimées dans l'ordre de réception).

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). Le traitement en mode multi-cœur ne peut pas garantir l'ordre d'impression par défaut .

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 l'option **Autre**.
4. Dans la section **Commentaires du moteur d'impression,** activer l'option **Impression supervisée**.

8.5 Retour D'information Sur Le Travail D'impression

CONSEIL: Les fonctionnalités de cet élément ne sont pas disponibles dans tous les produits NiceLabel Automation.

L'application qui fournit les données d'impression au NiceLabel Automation peut s'attendre à recevoir des informations concernant l'état des tâches d'impression. Le retour peut être simple tel que "Tout OK" le travail d'impression est réussie, ou plus détaillé en cas de problème. Pour des raisons de performances, par défaut NiceLabel Automation désactive les possibilités de renvoi 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 seront 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, tel que l'état des imprimantes réseau, le nombre de tâches 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 renvoi 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.

Le retour d'information peut se faire de deux façons.

Le déclencheur renvoie les informations concernant l'état des tâches 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 de la tâche d'impression. Le client peut envoyer les données au déclencheur, garder la connexion ouverte et attendre le retour. Pour utiliser cette méthode, il faut que le déclencheur la supporte.

Quand 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 renvoi d'informations. Le document WSDL (Web Service Description Language) décrit les détails concernant l'interface Service Web et 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 la tâche d'impression (données binaires).
- Déclencheur Serveur HTTP. Par défaut, ce déclencheur supporte le renvoi d'informations. NiceLabel Automation va utiliser le code de réponse HTTP standard pour indiquer l'état de la tâche 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 la tâche d'impression (données binaires).
- Déclencheur Serveur TCP/IP. Ce déclencheur supporte le renvoi d'information, mais pas automatiquement. Il faut donc 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 avoir le paramètre **Répondre à l'expéditeur.** 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 renvoi 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. Toute autre action de connectivité du trafic sortant peut être configurée comme ,,, etc. cette action sera placée sous l'action.

Pour renvoyer des informations sur un statut spécifique, tel que "erreur survenue", utiliser une des méthodes suivantes.

- **Utiliser une condition sur 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, pour utiliser l'action **Requête HTTP** après l'impression et envoyer les informations uniquement quand une erreur survient. Il faudra procéder ainsi:
    1. Ouvrir les propriétés du déclencheur.
    2. Dans le groupe de ruban Variable , cliquer sur le bouton **Variables internes** et activer la variable `LastActionErrorID`.
    3. Aller sur l'onglet Actions.

4. Ajouter l'action **Envoyer les Données au HTTP**.
5. Dans les propriétés de l'action, développer **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.

```
LastErrorActionID > 0
```

7. il faut également activer l'option **Ignorer l'échec** pour chaque action qui pourrait échouer. Cela préviendra Automation de ne pas arrêter complètement l'exécution des actions, mais de continuer par l'action suivante dans le même niveau hiérarchique.

> NOTE: Pour plus d'informations, consulter l'article Traitement d'Erreur.

- **Utiliser l'action Essayer.** L'action Essayer élimine la nécessité d'encoder les conditions. L'action fournit deux espaces réservés. L'espace réservé **Do** (faire) va contenir les actions à exécuter. Si une erreur survient, l'exécution s'interrompt et les actions dans l'espace réservé **Pour erreur** seront 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 le chapitre

8.6 Utiliser Le Mode D'impression Stocker/Rappeler

Le mode d'impression Stocker/Rappeler optimise le processus d'impression. Il augmente les capacités de réception de l'imprimante en diminuant le nombre de données à envoyer en cas de 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. Le NiceLabel Automation n'envoie que les données pour les objets variables des étiquettes et pour 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. Le stockage se fait depuis l'éditeur ou avec l'action NiceLabel Automation.

NOTE: Définir d'abord dans l'es propriétés de l'étiquette le mode d'impression Stocker et Rappeler pour pouvoir la stocker 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.

ATTENTION : 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 et Rappeler. Le pilote de l'imprimante pour laquelle l'étiquette a été créée dans l'éditeur d'étiquettes doit également être installé sur la machine sur laquelle NiceLabel Automation tourne.

8.7 Cluster Haute Disponibilité (failover)

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Enterprise**.

NiceLabel Automation peut utiliser le cluster haute-disponibilité de Microsoft (fail-over). 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 client qui envoient des données se connectent à l'adresse IP appartenant au cluster, pas à l'adresse IP du nœud.

Pour activer NiceLabel Automation en Haute-disponibilité, effectuer les opérations suivantes :

- Activer la fonctionnalité Microsoft Failover Clustering sur vos Serveurs Windows.
- Installer NiceLabel Automation sur chaque nœud.

- Activer la possibilité de basculer le cluster dans les propriétés de NiceLabel Automation sur chaque nœud.

  Suivre les instructions :

  1. Ouvrir **Fichier >Outils>Options**
  2. Sélectionner la section **Mode cluster**.
  3. Activer **le mode cluster.**
  4. 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 copiées dans ce dossier.

- Configurer le cluster pour démarrer NiceLabel Automation sur le second nœud quand le nœud maître est en panne.

8.8 Cluster De Répartition Des Charges

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation 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 groupés (appelés nœuds) sont raccordés physiquement par des câbles et par un logiciel. Les requêtes entrantes d'impression d'étiquettes sont distribuées entre tous les nœuds du cluster. Les clients qui envoient des données se connecteront à l'adresse IP appartenant au cluster, pas à l'adresse IP du nœud.

NOTE: Les déclencheurs TCP/IP sont utilisables avec un cluster de répartition des charges, il s'agit du Déclencheur Serveur TCP/IP, du Déclencheur Serveur HTTP et du Déclencheur Web Service.

Pour activer la répartition des charges du NiceLabel Automation, effectuer les opérations suivantes :

- Activer la fonctionnalité Microsoft Load-balancing Clustering dans vos Serveurs Windows.
- Installer NiceLabel Automation sur chaque nœud.
- Charger les même fichiers de configuration dans Automation Manager sur chaque nœud.

9 Comprendre les structures de données

9.1 Comprendre Les Structures De Données

Ce chapitre démontre la structure de donnée basique qui est fréquemment utilisée dans les scénarios d'automation. Il faut lire les structures, extraire les valeurs intéressantes et les imprimer sur l'étiquette. Chaque exemple mentionné est utilisé dans les exemples installés avec le logiciel. Plus plus d'informations, consulter l'article Exemples.

- Base de données Texte
- CSV Composé
- Fichiers Binaires
- Données existantes
- Fichiers de Commande
- Données XML

9.2 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). Le Filtre de données non-structurées autorise les caractères binaires. Les caractères binaires permettent de définir des positions de champs, ils sont aussi utilisables comme valeurs de champs.

Exemple typique: l'exportation de données d'un système existant, sur lequel les données d'étiquettes sont délimitées par un caractère `<FF>` - saut de page.

Exemple
Dans ce cas, le déclencheur 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
 1e42055007500500001001019
 1322000001502859
 W
 E
<FF><ESC>%-12345X<ESC>%-12345X@PJL USTATUSOFF
<ESC>%-12345X
```

Pour plus d'informations, consulter l'article Exemples.

9.3 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, voir les articles Caractéristiques des fichiers de commande, Caractéristiques Oracle XML et Caractéristiques SAP All XML.

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 poids="1,0 kg"
PRINTER "CAB A3 203DPI"
PRINT 1
```

Plus plus d'informations, consulter l'article Exemples.

9.4 CSV Composé

Un fichier composé CSV est un fichier texte contenant une structure CSV ainsi qu'une entête multi-ligne dans une autre structure. Le contenu ne peut pas être analysé par un seul filtre. Il faut configurer deux filtres, le premier un Filtre de Texte Structuré pour les champs de la section CSV et un second Filtre de données non-structurées pour les champs de la section d'entête. Il faut aussi définir deux actions Utiliser un Filtre de Données et exécuter les deux filtres sur les données reçues.

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"

Plus plus d'informations, consulter l'article Exemples.

9.5 Données Existantes

Les données existantes sont une exportation non-structurée ou semi-structurée des applications existantes. Ces données n'ont pas une structure CSV ou XML, il faut donc utiliser le Filtre de données non-structurées pour définir les positions des champs. Le filtre va extraire les valeurs des champs pour les imprimer sur les étiquettes.

Exemple
Il n'y a pas de règles de structure. Chaque champ doit être configuré manuellement.

HAWLEY ANNIE ER12345678 ABC XYZ
9876543210
PRE OP 07/11/12 **F 27/06/47** Hopital St Georges

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

Dépistage Anti-corps : Négatif
Stockage de l'échantillon :
ÉCHANTILLON VALABLE 24 HEURES, PAS D'HISTORIQUE DE TRANSFUSION FOURNI

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

Plus plus d'informations, consulter l'article Exemples.

9.6 Base De Données Texte

La base de donnée 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, voir l'article Activer la Structure Dynamique.

Exemple

- **Fichier à champs délimités.** La première ligne du fichier peut contenir 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 à largeur de champs fixe.**

```
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
```

Plus plus d'informations, consulter l'article Exemples.

9.7 Données XML

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

XML signifie eXtensible Markup Language. Les balises XML ne sont pas prédéfinies, chacun peut librement définir ses propres balises pour décrire ses 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).

Exemples

Oracle XML

Le traitement d'Oracle XML est intégré dans le logiciel. Pas besoin de configurer de filtres pour extraire les données, il suffit de lancer l'action intégrée. Pour plus d'information sur la structure XML, consulter le chapitre Caractéristiques Oracle 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, il faut définir le filtre XML et définir les règles d'extraction de données. Pour plus de renseignements, consulter l'article Comprendre les Filtres.

```
<?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 XML est intégré dans le logiciel NiceLabel. Pas besoin de configurer de filtres pour extraire les données, il suffit de lancer l'action intégrée. Pour plus d'information sur la structure XML, consulter le chapitre 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>
```

Pour plus d'informations pratiques sur l'utilisation des données XML, consulter l'article Exemples.

10 Référence et résolution de problèmes

10.1 Types De Fichiers De Commande

10.1.1 Caractéristiques Des Fichiers De Commande

Les fichiers de commande contiennent les instructions pour le processus d'impression 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 :

10.1.2 Fichier De Commande CSV

Les commandes disponibles dans les fichiers de commande CSV sont un sous-ensemble des commandes NiceLabel. Commandes utilisables: **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.

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
```

Spécifications des commandes 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, et ils seront extraits avec leurs valeurs comme paires `nom-valeur`.

- **@Label.** Spécifie le nom de l'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 le chapitre du guide utilisateur de NiceLabel Automation. 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 définit l'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 sera enregistré dans la variable ayant le même nom avec sa valeur.

10.1.3 Fichier De Commande JOB

Le fichier de commande JOB est un fichier texte contenant les commandes NiceLabel. Les commandes sont exécutées dans l'ordre, de haut en 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.

Exemple De Fichier De Commande JOB

Ce fichier JOB va ouvrir l'étiquette `label2.nlbl`, paramétrer les variables et imprimer une étiquette. Comme aucune commande PRINTER n'est utilisée pour rediriger l'impression, l'étiquette sera imprimée en utilisant l'imprimante définie dans l'étiquette.

```
LABEL "label2.nlbl"
SET code="12345"
SET article="FUSILLI"
SET ean="383860026501"
```

```
SET poids="1,0 kg"
PRINT 1
```

10.1.4 Fichier De Commande XML

Les commandes disponibles dans les fichiers de commande XML sont un sous-ensemble des commandes NiceLabel. Utiliser les commandes suivantes : **LOGIN**, **LABEL**, **SET**, **PORT**, **PRINTER**, **SESSIONEND**, **SESSIONSTART** et **SESSIONPRINT**. La syntaxe change légèrement quand elle est utilisée dans un fichier XML.

L'élément de base du fichier de commande XML est `<Nice_Commands>`. L'élément qui doit suivre est `<Label>`, il 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.

Exemple De Fichier De Commande XML
L'exemple présente une vue structurelle des éléments et leurs attributs tels qu'ils peuvent être utilisés dans le 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>
```

Caractéristiques des commandes 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 pré-définies, d'autres peuvent avoir des valeurs personnalisées.

- **<Nice_Commands>.** C'est un élément de base.
- **<Label>.** Spécifie le fichier d'étiquette à ouvrir. Si l'étiquette est déjà ouverte, elle ne sera pas rouverte. 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 présenter plusieurs fois dans le fichier de commande.

- **Name.** Cet attribut contient le nom d'étiquette. Mettre éventuellement le nom du chemin. Obligatoire.

- **<Print_job>.** L'élément qui contient les données d'un travail d'impression. Cet élément peut se présenter plusieurs fois dans le fichier de commande.
  - **Printer.** A 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.
  - **Quantity.** Utiliser cet attribut 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.
  - **Skip.** A utiliser pour spécifier le nombre d'étiquettes à sauter au début. Fonctionnalité 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 Facultatif.
  - **Job_name.** A 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 Facultatif.
  - **Print_to_file.** Utiliser cet attribut pour spécifier le nom du fichier dans lequel il faut enregistrer les commandes de l'imprimante. Pour plus d'informations, voir l'article Rediriger l'impression vers un fichier. Optionnel.
  - **Identical_Copies**. utiliser cet attribut pour spécifier le nombre de copies à imprimer pour chaque étiquette. Optionnel. Optionnel.
- **<Session_Print_Job>.** Elément qui contient les données et les commandes pour une ou plusieurs sessions. L'élément peut contenir une ou plusieurs `<Session>`. Il envisage 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.
  - **Quantity.** A utiliser pour spécifier le nombre d'étiquettes à imprimer. Valeurs possibles : valeur numérique, chaîne de caractères VARIABLE ou chaîne ILLIMITEE. Pour plus d'informations, consulter l'article Obligatoire.
- **<Variable>.** L'élément qui assigne les valeurs aux variables de l'étiquette. Cet élément peut se présenter plusieurs fois dans le fichier de commande.
  - **Name.** L'attribut qui contient le nom de variable. Obligatoire.

Définition du Schéma 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: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>
```

10.1.5 Caractéristiques Oracle XML

Oracle a défini un format XML pour faciliter la compréhension, l'analyse et ensuite l'impression sur des étiquettes d'un contenu en XML. La définition du type de document XML (DTD) permet de définir les balises à utiliser dans le fichier XML. Oracle va générer les fichiers XML selon cette DTD et le logic[ie]l tiers va traduire l'XML selon cette DTD.

Pour exécuter un tel fichier de commande, utiliser l'action Exécuter le Fichier de Commande XML.

XML DTD

Ci-dessous, la DTD XML utilisée pour former la XML pour les formats XML synchrone et asynchrone. Elle définit les éléments qui seront 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>
<!ATTLIST label _PRINTERNAME CDATA #IMPLIED>
<!ELEMENT variable (#PCDATA)>
<!ATTLIST variable name CDATA #IMPLIED>
```

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

L'étiquette s'imprimera en `1` copie, sous le nom de travail d'impression `Serial` dans le spouleur. Le nom d'imprimante n'est pas spécifié dans le fichier XML, donc l'étiquette s'imprime sur l'imprimante définie dans le masque d'étiquette.

10.1.6 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.

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 variable	Valeur de variable
EPC	00037000657330
EPC_TYPE	SGTIN-96
EPC	urn:autoid:tag:sgtin:3.5.0037000.065774.8
PRODUIT	Produit
PRODUCT_DESCRIPTION	Description

L'étiquette s'imprimera en `1` copie, avec le nom du travail du spouleur `Writer_Device2004092916574`. Le nom d'imprimante n'est pas spécifié dans le fichier XML, donc l'étiquette s'imprime sur l'imprimante définie dans le masque d'étiquette.

10.2 Commandes Personnalisées

10.2.1 Utiliser Des Commandes Personnalisées

Les commandes NiceLabel sont utilisées dans les fichiers de commande pour contrôler l'impression des étiquettes. NiceLabel Automation exécute la commande du fichier de commande, de haut en bas. Pour plus d'informations, voir l'article Spécifications des Fichiers de Commande.

Ces commandes personnalisées spécifiques sont utilisées comme les actions dans le produit NiceLabel Automation.

NOTE: Par exemple, la commande **SETPRINTPARAM** s'utilise avec l'action **Configurer le Paramètre d'Impression** (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 en 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 avec l'action.

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 de renseignements, consulter l'article Utiliser le mode d'impression Stocker/Rappeler.

IGNOREERROR

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

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 [, printerName]
```

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 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 pour le nom d'imprimante fourni n'est pas installé ou pas disponible, la commande signalera une erreur.

MESSAGEBOX

```
MESSAGEBOX <message> [,<caption>]
```

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 réseau partagées.

Le paramètre `APPEND` est facultatif. Par défaut, le fichier est écrasé. Utiliser ce paramètre pour joindre des données à un fichier existant.

A 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 souvent 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.

- **Quantity.** Spécifie le nombre d'étiquettes à imprimer.
  - **<nombre>**} Le nombre d'étiquettes spécifié sera imprimé.
  - **VARIABLE.** Spécifie qu'une variable d'étiquette est définie comme *quantité variable* et contiendra le nombre d'étiquettes à imprimer. L'étiquette va déterminer le nombre d'étiquettes à imprimer.
  - **UNLIMITED.** Avec une base de données comportant les valeurs des objets, l'impression illimité imprimera autant d'étiquettes qu'il y a de données dans la base de données. Sans base de données, le nombre maximal d'étiquettes supporté par l'imprimante thermique sera imprimé.
- **Skip.** 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.

- **Copies Identiques d'étiquettes** Spécifie le nombre d'étiquettes identiques qui doivent être imprimées.
- **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 la ligne d'état de l'éditeur d'étiquette. 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 de la tâche d'impression visible dans le Spouleur Windows. Si le nom contient des caractères d'espacement ou virgules, il faut le 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. Pour 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 **SESSIONSTART** (début de session).

- **Quantity.** Spécifie le nombre d'étiquettes à imprimer.
- **Skip.** 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 - début de session

```
SESSIONSTART - début de session
```

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 une tâche d'impression différente. Pour joindre les données d'étiquettes dans un flux d'impression, il faut utiliser les commandes d'impression de session. Il faut commencer par la commande SESSIONSTART, suivie par un nombre quelconque de commandes SESSIONPRINT et terminer par la commande SESSIONEND.

Utiliser ces commandes pour optimiser le processus d'impression des étiquettes. L'impression d'étiquettes provenant dans un seul flux d'impression est plus rapide que l'impression d'étiquettes dans différents travaux d'impressions.

Il faut respecter certaines règles pour éviter de rompre la session d'impression.

- 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 compteurs. 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, il faut 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 Charriot) 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 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 Missing variable reference dans le guide utilisateur.

  NOTE: La chaîne de caractères codée en Base64 doit être fournie entre guillemets (").

TEXTQUALIFIER Délimiteur de texte

```
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, parfois il est 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"%
```

10.3 Accès Aux Ressources Réseau Partagées

Cet article définit les recommandations à suivre lors de l'utilisation de ressources de réseau partagées.

- **Privilèges 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 de noms pour spécifier les disques réseau. NiceLabel Automation va automatiquement remplacer la syntaxe avec le lien sur le disque par la syntaxe UNC.

  EXEMPLE: Quand l'adresse du fichier est `G:\Labels\label.nlbl`, la présenter sous la forme UNC `\\server\share\Labels\label.nlbl` (sans laquelle le lecteur G: drive est lié à `\\server\share`).

- **Notation pour accéder au fichiers dans Control Center.** Pour ouvrir le fichier du Stockage de Documents dans le Control Center, utiliser la notation HTTP: `http://-servername:8080/label.lbl`, ou la notation WebDAV: `\\servername@8080\DavWWWRoot\label.lbl`.

  Notes additionnelles :

  - Le compte utilisateur qui exécute le service NiceLabel Automation sera utilisé pour récupérer les fichiers du Stockage de Documents. Cet utilisateur doit être paramétré dans la configuration du Control Center pour pouvoir accéder 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 document est utilisable avec les produits **NiceLabel LMS Enterprise** et **NiceLabel 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 KB 265.

10.4 Stockage De Documents Et Contrôle Des Versions Des Fichiers De Configuration

Le Stockage de document 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 leur 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 document:

- **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 révision 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.

  ATTENTION : 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 du NiceLabel Control Center connecté.

10.5 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 tournera toujours sous le nombre de bits du système Windows.

Windows 32 Bits
Avec Windows 32 bits, installer uniquement des pilotes de base de données 32 bits. Le même pilote de base de données sera utilisé pour configurer le déclencheur dans le 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.

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 utiliseront les pilotes de base de données 64 bits. Les applications exécutées en 32 bits utiliseront les pilotes de base de données 32 bits.

Par défaut le service Automation tourne en 64 bits. Il utilisera donc les pilotes de base de données 64 bits pour se connecter aux bases de données. Quand 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 **Proxy** NiceLabel qui tourne toujours en 32 bits.

10.6 Remplacement Automatique De La Police

Il est possible de créer des masques d'étiquettes pour imprimer des objets texte formatés en polices imprimante, Toutefois, lors de l'impression d'une telle étiquette sur une autre imprimante, il se peut que les polices sélectionnées ne soient pas disponibles sur la nouvelle imprimante. La nouvelle imprimante supporte probablement un jeu de polices internes complètement différent. Les polices peuvent être semblables mais disponibles sous un autre nom.

Un problème similaire peut se présenter si la police Truetype utilisée dans l'étiquette n'est pas installée sur la machine utilisée par NiceLabel Automation pour imprimer les étiquettes.

NiceLabel Automation peut être configuré pour remplacer automatiquement les polices utilisées dans l'étiquette par des polices compatibles. Le lien entre les polices données et les polices de remplacement peut être effectué en fonction des noms de police. Si la police originale est introuvable, NiceLabel Automation essayera d'utiliser la première police de remplacement disponible, comme définie dans la table de mappage. Si aucune police de remplacement n'est trouvée, la police Arial Truetype sera utilisée.

NOTE: Si une police de remplacement est configurée, les règles de mappage seront exécutées quand l'imprimante change sur l'étiquette.

ATTENTION : La configuration du remplacement de la police n'est pas préservée lors de la mise à jour du logiciel. Veiller à faire une sauvegarde avant la mise à jour.

Configuration du Mappage de Police

Effectuer les opérations suivantes pour configurer la police personnalisée :

1. Ouvrir l'explorateur de fichiers et rechercher le dossier suivant :

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

2. Ouvrir le fichier **fontmapping.def** dans un éditeur de texte XML.
3. Dans l'élément **FontMappings**, créer un nouvel élément avec un nom personnalisé.
4. Dans le nouvel élément, créer au moins deux éléments portant le nom **Mappage**.
    - La valeur du premier élément Mappage doit contenir le nom de la police originale.
    - La valeur du deuxième élément Mappage doit contenir le nom de la police de remplacement.

    > NOTE: Il peut y avoir des éléments de Mappage additionnels avec des nouveaux noms de polices. Si la première police de remplacement n'est pas disponible, NiceLabel Automation essaiera la suivante. Si aucune police de remplacement n'est disponible, Arial Truetype sera utilisée.

Exemple de configuration de Mappage

Dans cet exemple, 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. Si cette police n'est pas disponible, **Arial** Truetype sera utilisé.
- 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>
```

10.7 Changer Les Paramètres Par Défaut D'Impressions Multi Threads

CONSEIL: La fonctionnalité de cet élément est disponible dans **NiceLabel Automation Pro** et **NiceLabel Automation Enterprise**.

Chaque produit NiceLabel Automation peut tirer avantage des multiples cœurs du processeur. Chaque cœur sera utilisé pour exécuter un processus d'impression. 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 2017\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 mettra automatiquement à jour le service avec le nouveau nombre de threads d'impression.

10.8 Compatibilité Avec Les Produits NiceWatch

NiceLabel Automation peut charger les configurations qui ont été définies dans un des 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'étiquette. 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, la notification concernera :

- **La compatibilité avec la configuration du déclencheur.** A 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. Certaine sont totalement indisponibles, d'autres sont configurées différemment. Si le fichier MIS contient des fonctionnalités incompatibles, elles figureront dans une liste et seront supprimées 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 de NiceLabel Automation pour recréer la configuration.

- **La compatibilité avec le masque de l'étiquette.** Si les masques d'étiquettes existants contiennent des fonctionnalités non supportées par le moteur d'impression de NiceLabel Automation, il y aura des messages d'erreur dans le panneau du 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 la Base de connaissances - article KB251.

Ouvrir la configuration NiceWatch pour l'éditer

Ouvrir la configuration NiceWatch existante (fichier .MIS) dans Automation Builder et l'éditer dans Automation Builder. La configuration peut seulement être enregistrée au format .MISX.

Pour éditer la configuration NiceWatch, effectuer les opérations suivantes :

1. Démarrer Automation Builder.
2. Sélectionner **Fichier>Ouvrir fichier NiceWatch**.
3. Dans la boite de dialogue Ouvrir, rechercher le fichier de configuration NiceWatch (fichier .MIS).
4. Cliquer sur **OK**.

5. Si la configuration contient des fonctionnalités non-supportées, elle en affichera la liste. Elles seront retirées 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, il 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 l'interface 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 le bouton **Démarrer**.

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

10.9 Contrôler Le Service Avec Les Paramètres De Ligne De Commande

Ce chapitre fournit les informations pour démarrer et arrêter le Service d'Automation, pour contrôler les configurations à charger et connaître les déclencheurs qui sont actifs, depuis la ligne de commande.

NOTE: Il faut lancer le terminal de **Ligne de Commande** en mode administrateur. Cliquer à droite sur cmd.exe et sélectionner **Exécuter en tant qu'administrateur**.

Démarrer et arrêter les Services

Pour démarrer les deux services depuis la ligne de commande, utiliser les commandes suivantes :

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

Pour ouvrir le fichier de configuration quand le Service est démarré, utiliser :

```
net start NiceLabelAutomationService2017 [Configuration]
```

Par exemple :

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

Pour arrêter les services, utiliser les commandes suivantes :

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

Gérer les Configurations et Déclencheurs

Le service de NiceLabel Automation peut contrôler les paramètres de la ligne de commande Automation Manager. La syntaxe générale d'utilisation de paramètres dans la ligne de commande est la suivante :

```
NiceLabelAutomationManager.exe COMMAND Configuration [NomDéclencheur] [/SHOWUI]
```

NOTE: Note : Mettre le chemin complet vers le nom de la configuration, ne pas utiliser le nom de fichier seul.

Pour Ajouter (ADD) Une Configuration

La configuration fournie sera chargée dans le service. Aucun déclencheur ne démarrera. Avec le paramètre `/SHOWUI` inclus, Automation Manager démarrera.

```
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 Manager démarrera.

```
NiceLabelAutomationManager.exe RELOAD c:\Project\configuration.MISX /SHOWUI
```

Pour Enlever (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 les informations en retour de 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'état sont collectés dans la variable 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 NiceLabel LMS Enterprise ou NiceLabel 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.

10.10 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.

L'utilisateur peut configurer le service pour remplacer certaines parties de la chaîne de connexion pendant que le déclencheur s'exécute. Une instance d'Automation peut 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 au cours du développement et de les exécuter en production sans rien changer à la configuration.

La logique de remplacement de la chaîne de connexion est définie dans le fichier `DatabaseConnections.Config` du dossier système de Automation.

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

Le fichier de configuration XML définit des paires source-destination. L'élément `<Replacement>` contient un élément `<De>` et un élément `<A>` . Au cours de l'exécution du déclencheur, la chaîne "De" est remplacée par la chaîne "A". Il est possible de définir autant d'éléments de `<Remplacement>` que nécessaire.

Le fichier de configuration n'est pas installé avec Automation. Il faut l'ajouter en utilisant la structure donnée en exemple. Les mêmes règles de rechercher & remplacer s'appliqueront à tous les déclencheurs du service Automation sur cette machine.

NOTE: Veiller à redémarrer les deux Services Automation après avoir ajouter le fichier de configuration dans le dossier système de l'Automation.

Exemple

Le déclencheur existant contient une connexion au serveur Microsoft SQL `mySQLServer` et la base de données `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.

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

<DatabaseConnectionReplacements>

<Replacement>

<From>Data Source=mySQLServer</From>

<To>Data Source=NEW_mySQLServer</To>

</Replacement>

<Replacement>

<From>Initial Catalog=myDatabase</From>

<To>Initial Catalog=NEW_myDatabase</To>

</Replacement>

</DatabaseConnectionReplacements>
```

10.11 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 :

- **Entrer les caractères manuellement** en utilisant un des exemples de syntaxe décrits :
  - Utiliser la syntaxe `<caractère_spécial_acronyme>`, tel que `<FF>` pour FormFeed (saut de page), ou `<CR>` pour CarriageReturn (retour charriot), ou `<CR><LF>` pour une nouvelle ligne.
  - Utiliser la syntaxe `<#nombre>`, tel que `<#13>` pour CarriageReturn (retour charriot) ou `<#00>` pour le caractère nul.

  Pour plus d'informations, voir 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.

10.12 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

10.13 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 l'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 trouvent dans la solution.

- Imprimer l'étiquette
- Installer l'imprimante
- Envoyer les données à l'imprimante
- Aperçu de l'étiquette

- Paramétrer l'imprimante
- Configurer les paramètres d'impression

Chacune de ces action 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 un poste pour une imprimante spécifique en cochant la case **Conservée**. Cette imprimante sera toujours disponible.

ATTENTION : Si le nombre d'imprimantes défini par la licence est dépassé, le logiciel rentre 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 laisse le temps de résoudre le problème de licence sans rupture au niveau de l'impression ou de la conception d'étiquettes. C'est souvent dû au remplacement d'imprimantes, si les nouvelles imprimantes sont utilisées en même temps que les anciennes. 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.

CONSEIL: Pour en savoir plus sur le mode de licence de NiceLabel 2017, lire ce document.

10.14 Fonctionnement En Mode Service

NiceLabel Automation fonctionne comme un service Windows. Il est conçu pour ne pas exiger d'intervention durant le traitement de 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 privilèges 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 privilèges suffisant, 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 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 le service
- 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%`.
- 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 ce nouveau compte se connecte à l'ordinateur.
- 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 privilèges 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 que **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.

Pour plus d'informations, consulter l'article Accès aux ressources réseau partagées.

Mode Service : 32 bits vs 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 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 l'article 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\NiceLabelAutomationService2017
```

- Changer le nom de fichier en **%PROGRAMDATA%\NiceLabel\NiceLabel 2017\FileCache**, en conservant le chemin existant.
- Redémarrer le service NiceLabel Automation.

ATTENTION : Il n'est pas recommandé de changer le mode service NiceLabel Automation. en cas d'obligation, effectuer d'abord des tests étendus du déclencheur avant le déploiement dans l'environnement de production.

10.15 Ordre De Recherche Des Fichiers Requis

Quand NiceLabel Automation essaye de charger un fichier d'étiquette ou d'image, il tente de le localiser à différents endroits.

NiceLabel Automation essaiera de trouver le fichier dans l'ordre suivant :

1. Il vérifie si le fichier existe à l'endroit spécifié par l'action.
2. Il vérifie si le fichier existe dans le même dossier que le fichier de configuration (.MISX).
3. Il vérifie si le fichier d'étiquette existe dans le dossier .\Labels (pour les fichiers graphiques il vérifie le dossier .\Graphics).
4. Il vérifie si le fichier d'étiquette existe dans le dossier ..\Labels (pour les fichiers graphiques il vérifie le dossier ..\Graphics).
5. Vérifie si le fichier existe dans le dossier global d'étiquettes comme configuré dans les options.

Si le fichier n'existe dans aucun de ces endroits, l'action échoue et une erreur est signalée.

10.16 Sécuriser L'accès Aux Déclencheurs

Dans certains déploiements,il faut sécuriser l'accès 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 des sécurités d'accès au cours de leur conception. 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

Avec des déclencheurs TCP/IP, tels que Déclencheur Serveur TCP/IP, Déclencheur Serveur HTTP ou Déclencheur Web Service, il faut permettre aux applications externes de se connecter aux déclencheurs. Chaque déclencheur fonctionne dans le service NiceLabel Automation, pour lequel l'accès est gouverné par le Pare-feu Windows. Un pare-feu fonctionne comme la serrure de votre porte d'entrée - il aide à maintenir les intrus à l'extérieur.

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.
- 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 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. Quand 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. Quand 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 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 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.

10.17 Sessions D'impression

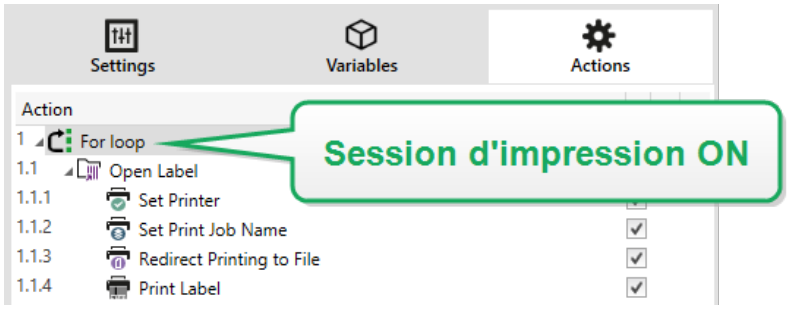
Une session d'impression permet d'imprimer plusieurs étiquettes dans un seul travail d'impression. Quand une session d'impression est activée, l'imprimante reçoit, traite et imprime d'un coup toutes les étiquettes du travail d'impression. Ainsi la vitesse d'impression augmente en raison du traitement continu du paquet d'étiquettes.

CONSEIL: 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 a 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 dans 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.

10.18 Conseils Et Astuces Pour Utiliser Des Variables Dans Les Actions

Suivre les recommandations ci-dessous pour utiliser des variables avec les actions de NiceLabel Automation.

- **Mettre les variables entre crochets.** Quand des variables contenant des espaces dans leur nom se réfèrent à des variables dans des actions, il faut mettre les variables entre crochets, ex: `[Nom Produit]`. Utiliser également les crochets si le nom de la variable est un nom réservé, par ex. 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 leDéclencheur 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.** Pour 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 Exécuter un Script 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).

10.19 Mode De Traçage

Par défaut, NiceLabel Automation enregistre les événements dans la base de données du journal. Ce journal comporte les informations importantes, telles que les informations rapportées après exécution des actions, après exécution des filtres et les informations concernant les mises à jour 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.

ATTENTION : Le mode de traçage va ralentir le traitement. Il ne faut l'utiliser que quand l'équipe de support NiceLabel le demande.

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 2017
```

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/Diagnostics/Tracing/Enabled` et lui assigner la valeur

VRAI.

Le fichier doit avoir le contenu suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<Configuration>
<Common>
<Diagnostics>
<Tracing>
<Enabled>Vrai</Enabled>
<Folder>c:\Troubleshooting\TracingLogs</Folder>
</Tracing>
</Diagnostics>
</Common>
...
</Configuration>
```

5. Lors de l'enregistrement du fichier, le service NiceLabel Automation va appliquer les paramètres automatiquement.
6. Pa défaut, les fichiers de traçage (*.LOG) vont apparaître dans le même dossier système.

   Changer le dossier en le spécifiant dans l'élément `Dossier`. Cet élément est facultatif.
7. Démarrer Automation Manager pour confirmer que le mode de traçage est activé. Il affichera le texte **Le traçage a été activé** dans l'interface de notification au-dessus de la liste de déclencheurs.

10.20 Comprendre Les Paramètres D'imprimante Et DEVMODE

NOTE: La structure de données DEVMODE fait partie de la Structure GDI d'Impression API de Windows. Ce sont des informations techniques complexes, utilisées seulement pour des requêtes très spécifiques.

A 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 l'étiquette.

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 sur l'une ou l'autre imprimante d'étiquette peut produire des résultats différents. Un même fichier d'étiquette peut demander des paramètres supplémentaires dans le pilote d'imprimante, tels que des réglages de marges, vitesse et température d'impression pour produire des résultats comparables. 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.

Définition de DEVMODE

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, tels 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 de données dans la partie publique dépend du développeur de pilote et se présente sous la forme d'un flux de données binaires.

Changer le DEVMODE

La structure de données 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. 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 d'imprimante par défaut.** Ils sont définis dans **Propriétés d'imprimante> Onglet Avancés> 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'affecteront que l'utilisateur actuel.
- **Paramètres d'étiquette spécifiques.** L'éditeur utilisant le logiciel NiceLabel NiceLabel peut choisir d'incorporer le DEVMODE dans l'étiquette elle-même. 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 d'impression personnalisés enregistrés dans l'étiquette** dans *Fichier>Paramètres Étiquette>Onglet Imprimante* de Designer Pro *.* Les paramètres d'imprimante sont modifiables dans l'étiquette en sélectionnant *Fichier>Paramètres Imprimante*.

Appliquer le DEVMODE personnalisé à l'impression

Dans NiceLabel Automation ouvrir un fichier d'étiquette et lui appliquer le DEVMODE

personnalisé. A l'impression de l'étiquette, le masque de l'étiquette est pris dans le fichier .LBL et le DEVMODE applique le formatage spécifique lié à l'imprimante. Comme cela il n'y a qu'une seule étiquette maître. L'impression sera la même, quelle que soit l'imprimante utilisée, car les paramètres optimaux d'impression sont appliqués pour cette imprimante.

Pour appliquer le DEVMODE à l'étiquette, il y a deux options :

1. Utiliser l'action définir le **Paramètre d'imprimante.**
2. Le fichier de commande JOB, plus spécialement la commande **SETPRINTPARAM** avec le paramètre **PRINTERSETTINGS**. Pour plus d'informations, voir Utiliser des commandes personnalisées.

10.21 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 est correct dans Automation Builder, le service peut rapporter un message d'erreur, en raison des différence d'information 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 les commandes `/user` pour spécifier le compte utilisateur et `/savecred` pour introduire le mot de passe une seule fois, il sera ainsi mémorisé pour la fois suivante.

11 Exemples

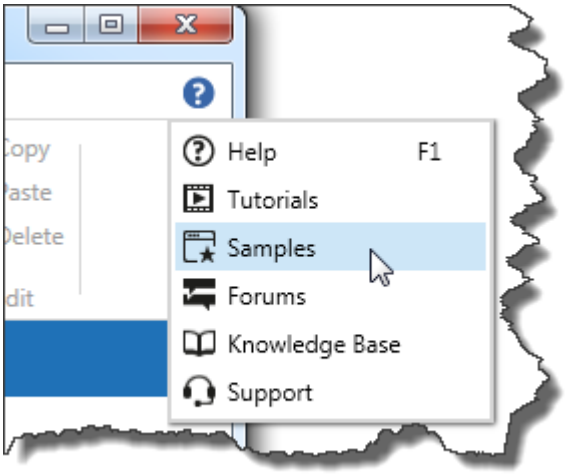
11.1 Exemples

NiceLabel Automation est livré avec des exemples qui décrivent les procédures de configuration pour les structures de données fréquemment utilisées et la configuration d'actions. Ils permettent de se familiariser avec la configuration de filtres pour extraire les données des fichiers CSV (Comma Separated Values), d'anciennes données exportées, de fichiers imprimante, de documents XML, de fichiers binaires, etc.

Un raccourci vers le dossier d'exemples est disponible dans Automation Builder.

Pour ouvrir le dossier d'exemples dans Windows Explorer, effectuer les opérations suivantes :

1. Ouvrir Automation Builder.
2. Cliquer sur le point d'interrogation dans le coin supérieur droit.
3. Sélectionner **Exemples**



4. Le dossier contenant les fichiers d'exemples s'ouvrira dans Windows Explorer.
5. Consulter le fichier **README.PDF** dans chaque dossier.

Les exemples sont installés dans le dossier suivant:

EXEMPLE: %PUBLIC%\Documents\NiceLabel 2017\Automation\Samples

qui sera transformé en
c:\Users\Public\Documents\NiceLabel Samples\Automation

12 Assistance technique

12.1 Assistance Technique En Ligne

Les dernières versions, mises à jour, solutions de contournement des problèmes et les Questions Fréquemment Posées (FAQ) se trouvent sur les pages du site web www.nicelabel.com.

Pour plus d'informations, se référer à :

- Base de connaissance: http://www.nicelabel.fr/support/knowledge-base
- Assistance NiceLabel : http://www.nicelabel.fr/support/technical-support
- Tutoriels NiceLabel : http://www.nicelabel.fr/learning-center/tutorials
- Forums NiceLabel: http://forums.nicelabel.com/

NOTE: Les titulaires d'un contrat de maintenance (SMA) doivent contacter le service d'assistance premium comme spécifié sur le contrat.

Americas

+1 262 784 2456

sales.americas@nicelabel.com

EMEA

+386 4280 5000

sales.@nicelabel.com

Germany

+49 6104 68 99 80

sales@nicelabel.de

China

+86 21 6249 0371

sales@nicelabel.cn

www.nicelabel.com