

Nalios EDI Connector

Guide complet d'utilisation — comprendre et configurer les échanges automatisés avec vos partenaires, sans écrire une ligne de code

Webhook

FTP / SFTP

AS2

XML · JSON · EDIFACT · Largeur fixe · CSV

- 
1. À quoi sert ce module ?
 2. Petit lexique EDI (à lire en premier)
 3. Se repérer dans le module
 4. Les canaux : comment les fichiers voyagent
 5. Les profils : la recette de lecture
 6. Les déclencheurs : automatiser l'envoi
 7. Suivre les échanges au quotidien
 8. Le protocole AS2 en détail
 9. Exemple complet de bout en bout
 10. Qui a le droit de faire quoi
 11. Questions fréquentes et dépannage
 12. Glossaire complet

1. À quoi sert ce module ?

Aujourd'hui, quand un client ou un fournisseur envoie une commande, une facture ou un bon de livraison, il ne l'envoie pas forcément en tapant à la main dans Odoo. Beaucoup de grandes entreprises (distributeurs, groupes industriels) échangent leurs documents commerciaux sous forme de **fichiers informatiques normalisés**, envoyés automatiquement d'un système à l'autre.

C'est ce qu'on appelle l'**EDI** — l'Échange de Données Informatisé. Le module **Nalios EDI Connector** est le "traducteur" qui reçoit ces fichiers, comprend leur contenu, et crée ou met à jour automatiquement les enregistrements correspondants dans Odoo (un client, une commande, une facture...) — sans qu'un humain ait besoin de ressaisir quoi que ce soit.

Il fait aussi l'inverse : il peut construire un fichier EDI à partir d'une donnée Odoo (par exemple confirmer une livraison) et l'envoyer automatiquement au partenaire.



💡 En une phrase

Le module regarde arriver un fichier, comprend ce qu'il contient grâce à des règles que vous avez configurées (pas du code), et transforme ça en commandes, clients, factures... directement dans Odoo. Et il peut faire la même chose en sortant.

Pourquoi c'est utile

- **Zéro ressaisie** : les commandes reçues d'un gros client arrivent directement dans Odoo, avec leurs lignes de produits déjà remplies.
- **Aucune ligne de code à écrire** : tout se configure dans l'interface Odoo (quel champ va où), même si le format du partenaire change avec le temps.
- **Traçabilité complète** : chaque fichier reçu ou envoyé est enregistré, avec son statut (réussi, en erreur, en attente) et peut être rejoué en cas de problème.
- **Plusieurs façons de transporter les fichiers** : par un lien web (webhook), un serveur de fichiers (FTP/SFTP), ou le protocole sécurisé du secteur de la distribution (AS2).

2. Petit lexique EDI

Avant d'aller plus loin, quelques mots reviendront tout le temps. Les comprendre maintenant rendra le reste du guide beaucoup plus simple.

Format du fichier : comment le contenu est écrit

Un même document (une commande, par exemple) peut être écrit de plusieurs façons différentes — un peu comme la même information peut être dite en français ou en anglais. Le module comprend cinq "langues" :

Format	À quoi ça ressemble	Utilisé par
XML	<pre><Commande><Ref>12345</Ref> </Commande></pre>	Beaucoup d'ERP et de sites e-commerce
JSON	<pre>{"commande": {"ref": "12345"}}</pre>	Applications web modernes, API
EDIFACT	<pre>BGM+220+12345+9'</pre>	Grande distribution, industrie (le standard historique du secteur)
Largeur fixe	<pre>HH05410000000001...</pre> (chaque champ occupe toujours les mêmes colonnes)	Anciens formats "mainframe" hérités de certains distributeurs, encore utilisés aujourd'hui
CSV	<pre>ref;client;produit;qte</pre>	Exports de tableurs, imports produits/stock

Le module sait lire et écrire les cinq — vous choisissez le bon format selon ce que votre partenaire utilise.

💡 Largeur fixe, c'est quoi exactement ?

Contrairement au XML ou au CSV, il n'y a **aucun séparateur** entre les champs : chaque information occupe toujours les mêmes colonnes, quelle que soit sa longueur (complétée par des espaces si besoin). Le fichier commence en général chaque ligne par un petit code qui dit "ceci est une ligne d'en-tête" ou "ceci est une ligne produit" — c'est ce code que le profil utilise pour savoir comment lire la ligne (voir section 5).

Canal : le moyen de transport

Le **canal**, c'est la façon dont le fichier voyage entre votre partenaire et Odoo. Ce n'est pas le contenu du fichier, c'est le "camion" qui le transporte :

- **Webhook** : le partenaire (ou son système) appelle une adresse web pour déposer le fichier, comme on soumet un formulaire en ligne.

- **FTP / SFTP** : un dossier partagé où les fichiers sont déposés, qu'Odoo va consulter régulièrement (comme une boîte aux lettres qu'on relève).
- **AS2** : un protocole spécial, très utilisé par les grands distributeurs, qui signe et chiffre les fichiers et garantit une preuve de réception (voir section 8).
- **Manuel** : le fichier est simplement stocké comme pièce jointe dans Odoo, sans être envoyé nulle part.

Profil : la recette de lecture

Une fois le fichier arrivé (peu importe le canal), il faut savoir **quoi faire avec son contenu**. C'est le rôle du **profil** : il dit "ce fichier est au format EDIFACT, il représente des commandes, et voici comment chaque champ du fichier correspond à un champ dans Odoo".

Mapping : la correspondance champ par champ

Un **mapping**, c'est une seule ligne de correspondance : "dans le fichier, la référence de commande se trouve ici → mets-la dans le champ Référence client de la commande Odoo". Un profil contient autant de mappings que de champs à récupérer.

À retenir

Canal = comment le fichier voyage. **Profil** = comment on le comprend. Un même profil peut être utilisé sur plusieurs canaux, et un canal peut servir à plusieurs profils.

3. Se repérer dans le module

Le module apparaît comme une application à part dans Odoo, avec sa propre icône.



L'application EDI dans l'écran d'accueil des applications Odoo

Une fois ouverte, elle propose trois menus :

- **Échanges** — la liste de tous les fichiers reçus ou envoyés, avec leur statut. C'est l'écran que vous consulterez le plus souvent.
- **Déclencheurs** — permet d'envoyer automatiquement un fichier dès qu'un enregistrement Odoo change (par exemple, dès qu'une commande est confirmée).
- **Configuration** — contient les **Canaux** et les **Profils**, c'est-à-dire tout ce qu'il faut paramétrer avant que les échanges puissent commencer.

EDIÉchangesDéclencheursConfiguration

Nouveau

Échanges EDI⚙

Q Rechercher...

1-5 / 5<>⌵⌴⌲

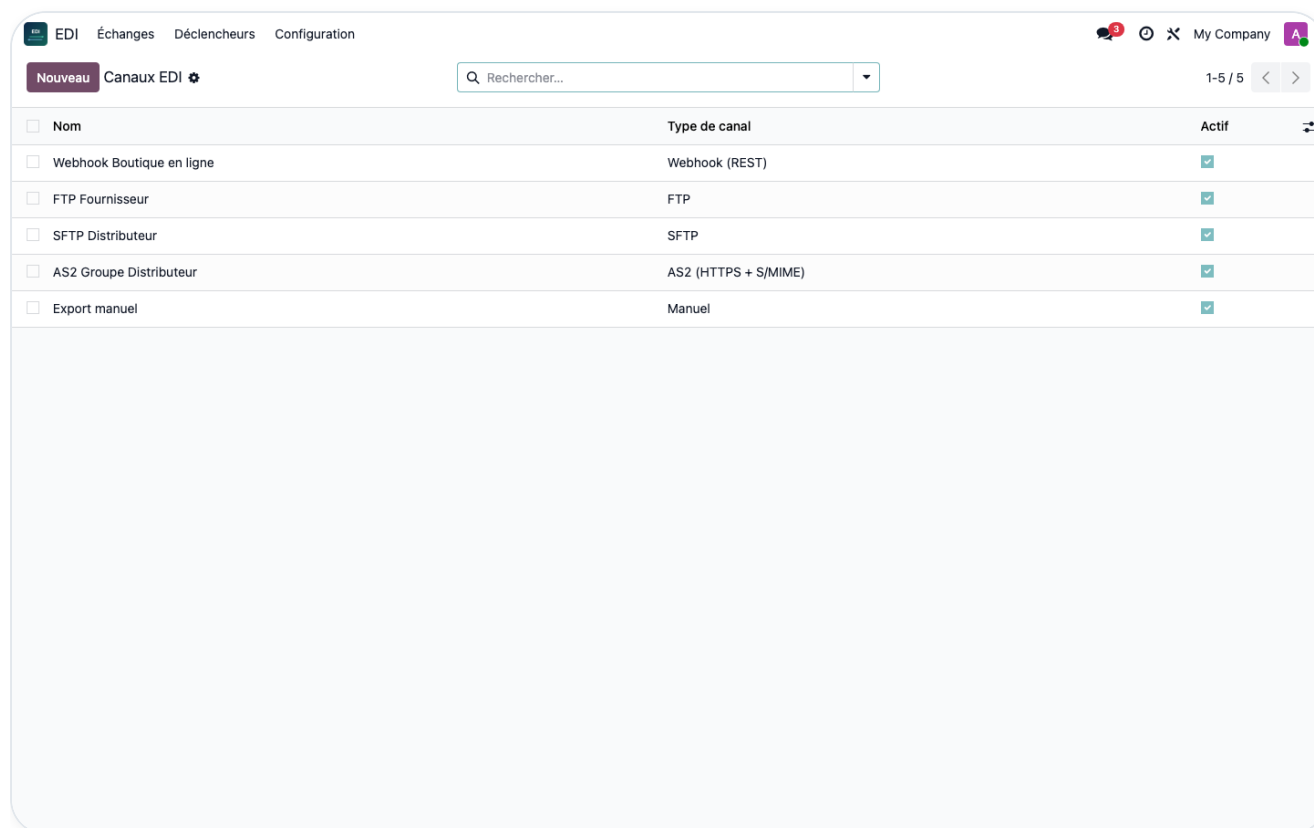
My CompanyA

☐	Nom	Date	Profil	Canal	Direction	Records	État	⌵
☐	EDI/2026/00005	13 août, 11:20	Commandes partenaire (exemple ORDERS)	AS2 Groupe Distributeur	Sortant	0	Succès	
☐	EDI/2026/00004	13 août, 11:20	Fiches partenaires (webhook)	SFTP Distributeur	Entrant	0	En attente	
☐	EDI/2026/00003	13 août, 11:20	Fiches partenaires (webhook)	FTP Fournisseur	Entrant	0	Erreur	
☐	EDI/2026/00002	13 août, 11:20	Fiches partenaires (webhook)	Webhook Boutique en ligne	Entrant	0	Succès	
☐	EDI/2026/00001	13 août, 11:20	Commandes partenaire (exemple ORDERS)	AS2 Groupe Distributeur	Entrant	1	Succès	

Les trois menus de l'application EDI

4. Les canaux : comment les fichiers voyagent

On crée un canal par partenaire (ou par type d'échange avec un partenaire). La liste des canaux ressemble à ceci :



☐ Nom	Type de canal	Actif
☐ Webhook Boutique en ligne	Webhook (REST)	☒
☐ FTP Fournisseur	FTP	☒
☐ SFTP Distributeur	SFTP	☒
☐ AS2 Groupe Distributeur	AS2 (HTTPS + S/MIME)	☒
☐ Export manuel	Manuel	☒

Liste des canaux — un par partenaire ou par moyen de transport

Canal de type Webhook

Le webhook fonctionne dans les deux sens : Odoo peut **recevoir** un fichier sur une adresse web unique (générée automatiquement, à donner au partenaire), et peut **envoyer** un fichier vers l'adresse du partenaire.

EDI

Échanges

Déclencheurs

Configuration

Nouveau

Canaux EDI

Webhook Boutique en ligne

1 / 5

Nom

Webhook Boutique en ligne

Type de canal

Webhook (REST)

WEBHOOK ENTRANT (IN)

Token ?

f9f0bf89-1500-4144-9df1-2f7d2304f451

URL Webhook ?

http://localhost:8180/edi/receive/f9f0bf89-1500-4144-9df1-2f7d2304f451

Régénérer le token

WEBHOOK SORTANT (OUT)

URL distante ?

http://localhost:8199/mock

Timeout (s) ?

30

Tester la connexion

ENVELOPPE EDIFACT (OPTIONNEL)

Notre identifiant (UNB) ?

ex: 1234567000006 (GLN)

Qualifiant expéditeur ?

14

Identifiant partenaire (UNB) ?

ex: 9876543000005 (GLN)

Qualifiant destinataire ?

14

Syntaxe UNB ?

UNOC:3

Configuration d'un canal Webhook

- 1 Webhook entrant (IN) — pour RECEVOIR des fichiers : le Token est généré automatiquement, ne le remplissez jamais vous-même. L'URL Webhook qui en découle est l'adresse exacte à donner à votre partenaire pour qu'il vous envoie ses fichiers. « Régénérer le token » change cette adresse — à n'utiliser que si elle a fuité, et il faudra alors la redonner au partenaire.
- 2 Webhook sortant (OUT) — pour ENVOYER des fichiers : ces deux champs ne servent que si Odoo doit envoyer des fichiers, pas en recevoir. URL distante = l'adresse chez le partenaire (c'est lui qui vous la fournit, ce n'est pas à vous de l'inventer). Timeout = combien de temps attendre une réponse avant d'abandonner (30 secondes convient dans la plupart des cas).

Rien à récupérer, rien à aller chercher

Contrairement au FTP/SFTP, un webhook n'a ni planning ni bouton « Déclencher le polling » — il n'y a tout simplement rien à aller vérifier. Dès que le partenaire poste un fichier sur l'URL Webhook, Odoo le traite sur-le-champ, dans cette même requête, et le résultat est déjà visible dans **Échanges** au moment où la requête du partenaire se termine. La seule chose à préparer de votre côté, c'est de vous assurer que cette URL est bien joignable depuis internet (derrière un pare-feu ou un reverse proxy, elle doit être exposée).

Canal de type FTP / SFTP

Ici, le partenaire dépose ses fichiers dans un dossier sur un serveur. Odoo va vérifier ce dossier à intervalle régulier et récupérer les nouveaux fichiers — mais cette **récupération automatique est désactivée par défaut** et couvre tous les canaux FTP/SFTP actifs en une fois. Activez-la et réglez la fréquence dans **Réglages** → **EDI**.

Paramètres
Paramètres généraux
Utilisateurs & Sociétés

Enregistrer
Ignorer
Paramètres

Rechercher...

Paramètres généraux
Ventes
EDI
Facturation

AS2

☒ **Activer AS2**
Active le canal AS2 (HTTPS + S/MIME) dans les Canaux EDI.
Nécessite le paquet Python `pyas2lib` sur le serveur.

Récupération automatique

☒ **Activer la récupération automatique**
Vérifie périodiquement tous les canaux FTP/SFTP actifs à la recherche de nouveaux fichiers, sans attendre qu'on clique sur « Déclencher le polling ».

Vérifier toutes les12Heures

Réglages → EDI : activation de la récupération automatique, vérification toutes les 12 heures — deux fois par jour

EDI
Échanges
Déclencheurs
Configuration

Nouveau
Canaux EDI
FTP Fournisseur

2 / 5

Nom

FTP Fournisseur

Type de canal

FTP

CONNEXION FTP/SFTP

Hôte ?

ftp.example-supplier.com

Nom d'utilisateur ?

nalios_edi

Port ?

21

Mot de passe ?

.....

Timeout (s) ?

30

DOSSIERS

Chemin distant ?

/edi/in

Dossier archives ?

/edi/done

Tester la connexion

Déclencher le polling

FICHIERS SORTANTS

Template nom de fichier ?

{date}_(sequence)

ENVELOPPE EDIFACT (OPTIONNEL)

Notre identifiant (UNB) ?

ex: 5400164000000 (GLN)

Qualifiant expéditeur ?

14

Identifiant partenaire (UNB) ?

ex: 5400141000009 (GLN)

Qualifiant destinataire ?

14

Syntaxe UNB ?

UNOC:3

Configuration d'un canal FTP — dossier de dépôt et dossier d'archivage

Une fois un fichier traité, il est **déplacé automatiquement** vers le dossier d'archives (au lieu d'être supprimé), pour garder une trace côté serveur du partenaire aussi.

💡 Comment est-on sûr de ne jamais traiter deux fois le même fichier ?

Le principe est simple : dès qu'un fichier est récupéré et traité, il quitte immédiatement le dossier de dépôt (déplacé vers les archives, ou supprimé si le déplacement échoue). Le passage suivant ne le voit donc plus jamais — pas besoin de mémoriser quoi que ce soit. Seul cas particulier : si le partenaire renvoie un jour un fichier avec exactement le même nom, il sera traité comme un nouveau fichier (le module ne compare pas le contenu) — c'est pourquoi chaque fichier doit porter un nom ou une référence unique, comme c'est l'usage standard en EDI.

📌 Un seul réglage, tous les canaux

Il n'existe qu'une seule fréquence de récupération automatique pour tout le module — elle vérifie *tous* les canaux FTP/SFTP actifs lors du même passage, pas un planning par partenaire. Si le serveur d'un partenaire n'est joignable que la journée, réglez la fréquence en conséquence (ex: toutes les 12 heures) plutôt que de garder les 5 minutes par défaut. Pas un partenaire FTP/SFTP ? Rien à faire — Webhook et AS2 fonctionnent tous les deux en push (le partenaire envoie directement à Odoo, ou Odoo envoie directement chez lui), donc aucune récupération n'est nécessaire ni d'un côté ni de l'autre.

📌 Pas urgent ? « Déclencher le polling » fonctionne toujours

La récupération automatique est optionnelle. Qu'elle soit activée ou non, le bouton « **Déclencher le polling** » sur un canal va toujours vérifier immédiatement, à la demande.

Nommer les fichiers sortants

Pour FTP et SFTP, le champ « **Template nom de fichier** » (sous « Fichiers sortants ») contrôle comment les fichiers générés sont nommés. En plus des variables génériques `{sequence}`, `{date}`, `{model}` et `{direction}`, `{field:xxx}` réutilise la valeur du mapping d'en-tête dont `xxx` est le Source Path — utile quand le partenaire attend une convention de nommage précise construite à partir de la référence du document lui-même.

EDI

Échanges

Déclencheurs

Configuration

Nouveau

Canaux EDI

FTP Fournisseur

2 / 5

Nom

FTP Fournisseur

Type de canal

FTP

CONNEXION FTP/SFTP

Hôte

ftp.example-supplier.com

Nom d'utilisateur

nalios_edi

Port

21

Mot de passe

.....

Timeout (s)

30

DOSSIERS

Chemin distant

/edi/in

Dossier archives

/edi/done

FICHIERS SORTANTS

Template nom de fichier

{field:idHeader}

Tester la connexion

Déclencher le polling

ENVELOPPE EDIFACT (OPTIONNEL)

Notre identifiant (UNB)

ex: 5400164000000 (GLN)

Qualifiant expéditeur

14

Identifiant partenaire (UNB)

ex: 5400141000009 (GLN)

Qualifiant destinataire

14

Syntaxe UNB

UNOC:3

Template nom de fichier avec {field:idHeader} — réutilisation de la valeur mappée sur « idHeader » comme nom de fichier

Canal de type AS2

AS2 est plus riche que les autres : signature électronique, chiffrement, accusé de réception. La section 8 lui est entièrement consacrée. Voici néanmoins un aperçu des champs :

EDI Échanges Déclencheurs Configuration

Nouveau Canaux EDI AS2 Groupe Distributeur

4 / 5

Nom AS2 Groupe Distributeur

Type de canal AS2 (HTTPS + S/MIME)

AS2

1 Notre AS2 ID ? NALIOSCORP

AS2 ID Partenaire ? RETAILERGROUP

URL distante ? https://edi.example-

Timeout (s) ? 30

3 URL de réception http://loca...

2 Signer les messages ☒

Chiffrer les messages ? ☒

Mode MDN ? Synchronique

Valider la chaîne de certificats ? ☐

CERTIFICATS

Notre certificat (PEM) ?

Notre clé privée (PEM) ?

Mot de passe clé privée ?

Certificat partenaire (PEM) ?

ENVELOPPE EDIFACT (OPTIONNEL)

Notre identifiant (UNB) ? 54100000000000

Qualifiant 14

Configuration d'un canal AS2

- 1 Les identifiants AS2 : un code que vous choisissez pour vous représenter, et le code du partenaire — à convenir ensemble avant de commencer.
- 2 Signer / chiffrer les messages : à cocher pour sécuriser les échanges (recommandé, souvent imposé par le partenaire).
- 3 L'adresse à donner au partenaire pour qu'il vous envoie des fichiers en retour.

Bloc « Enveloppe EDIFACT (optionnel) »

Sur chaque canal (quel que soit son type), un bloc « **Enveloppe EDIFACT** » apparaît en bas de la fiche. Il ne concerne que les échanges au format EDIFACT, et seulement si votre partenaire exige une **enveloppe d'interchange** — l'équivalent d'une enveloppe postale autour du document lui-même, avec l'identité de l'expéditeur et du destinataire.

- **Notre identifiant / Identifiant partenaire** : généralement un GLN (l'identifiant international de chaque entreprise), à convenir avec le partenaire.
- **Qualifiants et Syntaxe** : des codes techniques qui précisent le type d'identifiant utilisé — la valeur par défaut convient dans la grande majorité des cas.

À retenir

Si ce bloc est laissé vide, les fichiers EDIFACT sortants sont générés **sans enveloppe** (juste le document lui-même) — ce qui convient à de nombreux partenaires. Ne le remplir que si le partenaire demande explicitement une enveloppe UNB/UNZ.

Tester une connexion

Pour les canaux FTP, SFTP et Webhook, un bouton « **Tester la connexion** » permet de vérifier tout de suite que les identifiants et l'adresse sont corrects, sans attendre le premier vrai fichier.

EDJ Échanges Déclencheurs Configuration

Nouveau Canaux EDI Webhook Boutique en ligne

1 / 5 < >

Nom Webhook Boutique en ligne

Type de canal Webhook (REST)

WEBHOOK ENTRANT (IN)

Token ? a948b3e4-337e-4fee-9d6d-14bec431951c

URL Webhook ? http://localhost:8180/edi/receive/a948b3e4-337e-4fee-9d6d-14bec431951c

Régénérer le token

WEBHOOK SORTANT (OUT)

URL distante ? https://api.example-shop.com/edi/receive

1 Timeout (s) ? 30

Tester la connexion

ENVELOPPE EDIFACT (OPTIONNEL)

Notre identifiant (UNB) ? ex: 5400164000000 (GLN)

Qualifiant expéditeur ? 14

Identifiant partenaire (UNB) ? ex: 5400141000009 (GLN)

Qualifiant destinataire ? 14

Syntaxe UNB ? UNOC:3

Le bouton « Tester la connexion », ici sur un canal Webhook — disponible aussi sur FTP et SFTP

Un doute sur un champ ? Le petit « ? » répond

Chaque champ un peu technique de ces écrans (Token, URL, identifiants EDIFACT, qualifiants...) a une explication complète accessible en survolant le petit point d'interrogation à côté de son nom — pas besoin de deviner ce qu'il faut y mettre.

5. Les profils : la recette de lecture

Chaque profil décrit un type de document (une commande, une fiche client...) dans un format donné, et la façon de le transformer en donnée Odoo.

EDI

Échanges

Déclencheurs

Configuration

My Company

Nouveau

Profils EDI

Q

Rechercher...

1-2 / 2

<

>

☐ Nom	Format	Direction	Modèle Odoo	Canal	Échanges	Actif	
☐ Commandes partenaire (exemple ORDERS)	EDIFACT	Entrant	Commande client	AS2 Groupe Distributeur	2	☒	
☐ Fiches partenaires (webhook)	XML	Entrant	Contact	Webhook Boutique en ligne	3	☒	

Liste des profils configurés

Onglet Configuration

On y définit le **format** (XML, JSON, EDIFACT, largeur fixe ou CSV), le **modèle Odoo** cible (quel type de fiche créer : client, commande...), le sens (entrant/sortant), et deux repères techniques : le **tag racine** et le **tag d'enregistrement**. Pour le CSV, un bloc « **Options CSV** » apparaît en plus pour choisir le délimiteur (virgule, point-virgule...) et si la première ligne contient les en-têtes de colonnes.

EDI

Échanges

Déclencheurs

Configuration

Nouveau

Profils EDI

Commandes partenaire (exemple ORDERS) ⚙️

Échanges

3

1 / 3

Nom

Commandes partenaire (exemple ORDERS)

Modèle Odoo

Commande client

Format

EDIFACT

Canal

AS2 Groupe Distributeur

Direction

Entrant

LIGNES IMBRIQUÉES (OPTIONNEL)

Modèle des lignes ?

Ligne de commande

Champ lignes (one2many)

Lignes de commande (Commande client)

Filtre sur les lignes ?

Faire correspondre **tous les enregistrements**

Nouvelle règle

→ 4 enregistrement(s)

Configuration

Mappings d'en-tête

Mappings de lignes

Règles d'action

Tag racine XML ?

ORDERS

Tag enregistrement ?

LIN

Onglet Configuration d'un profil — ici pour lire des commandes EDIFACT et créer des commandes de vente avec leurs lignes

💡 Tag racine et tag d'enregistrement : deux questions différentes

Tag racine répond à « quel TYPE de document est-ce, dans son ensemble ? ». Il ne sert que si un même canal peut recevoir plusieurs types de documents différents (parfois des commandes, parfois des factures) — le module s'en sert pour choisir automatiquement le bon profil. Si ce canal ne reçoit toujours qu'un seul type de document, ce champ peut rester vide.

Tag d'enregistrement répond à « où commence UNE ligne, à l'intérieur du fichier ? ». Celui-là est presque toujours utile : c'est ce qui dit au module « une nouvelle ligne démarre ici ».

Format	Tag racine (exemple)	Tag d'enregistrement (exemple)
XML	<code>Orders</code> (tag qui enveloppe tout le fichier)	<code>Order</code> (tag répété pour chaque ligne)
JSON	la clé de premier niveau	la clé qui contient la liste des lignes
EDIFACT	<code>ORDERS</code> (type de message après <code>UNH</code>)	<code>LIN</code> (segment qui démarre chaque ligne produit)
Largeur fixe	—	<code>DD</code> ou <code>CC</code> (code qui démarre chaque nouvelle ligne)

Exemple concret : un fichier EDIFACT de commandes avec 3 produits → tag racine = **ORDERS** (« c'est une commande »), tag d'enregistrement = **LIN** (« chaque segment LIN = un produit ») → le module crée 3 lignes de commande.

Taille de lot : regrouper plusieurs enregistrements dans un fichier

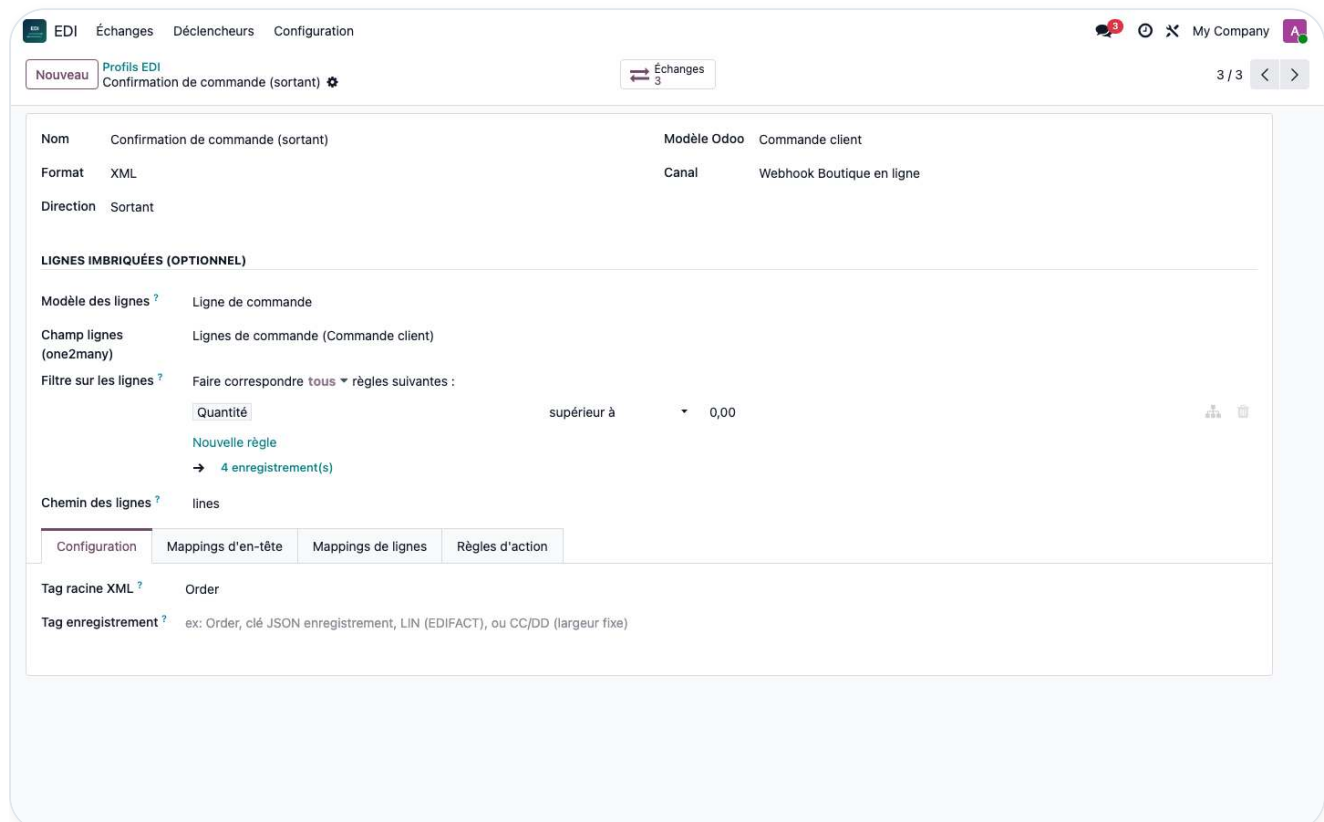
Quand plusieurs enregistrements sont envoyés ensemble en une fois (par exemple un export en masse sur un lot de commandes, plutôt qu'un déclencheur par commande), la « **Taille de lot** » contrôle combien d'entre eux atterrissent dans un même fichier généré : 0 signifie pas de limite — tout dans un seul fichier. Ce réglage ne s'applique qu'aux profils *sans* « Lignes imbriquées » — un profil imbriqué produit toujours un fichier par enregistrement parent.

Commande avec lignes : "Lignes imbriquées"

Une commande n'est pas juste un en-tête (le client, la référence) — elle a aussi des lignes de produits. Le bloc « **Lignes imbriquées** » permet d'indiquer que ce profil doit créer *une seule* commande avec *toutes ses lignes* attachées, plutôt qu'un enregistrement séparé par ligne de produit. On y indique le modèle des lignes (ex: Ligne de commande) et le champ qui les relie à l'en-tête.

Filtre sur les lignes : exclure certaines lignes

Une fois les « Lignes imbriquées » configurées, un « **Filtre sur les lignes** » apparaît : une condition optionnelle qui décide quelles lignes finissent réellement dans le fichier sortant. Les lignes qui ne correspondent pas sont simplement ignorées — pas besoin de nettoyer les données en amont.



The screenshot shows the Odoo EDI configuration interface for a profile named "Confirmation de commande (sortant)". The interface is divided into several sections:

- Header:** Includes navigation tabs (EDI, Échanges, Déclencheurs, Configuration), a "Nouveau" button, and a "Profils EDI" dropdown.
- Exchange Information:** Shows "Échanges 3" and "3 / 3" pages.
- General Settings:**
 - Nom:** Confirmation de commande (sortant)
 - Format:** XML
 - Direction:** Sortant
 - Modèle Odoo:** Commande client
 - Canal:** Webhook Boutique en ligne
- LIGNES IMBRIQUÉES (OPTIONNEL):**
 - Modèle des lignes:** Ligne de commande
 - Champ lignes (one2many):** Lignes de commande (Commande client)
 - Filtre sur les lignes:** Faire correspondre tous règles suivantes:
 - Quantité:** supérieur à 0,00
 - Nouvelle règle:** → 4 enregistrement(s)
 - Chemin des lignes:** lines
- Configuration Tabs:** Configuration, Mappings d'en-tête, Mappings de lignes, Règles d'action.
- Tag racine XML:** Order
- Tag enregistrement:** ex: Order, clé JSON enregistrement, LIN (EDIFACT), ou CC/DD (largeur fixe)

Le bloc « Lignes imbriquées » une fois configuré : Filtre sur les lignes (seules les lignes avec une quantité supérieure à 0) et Chemin des lignes (voir plus bas) côte à côte

💡 Un cas fréquent : ignorer les lignes vides ou non physiques

Une livraison peut contenir des lignes qui ne doivent pas arriver chez le partenaire : une ligne à quantité 0, un service (frais de manutention), une palette... Un filtre du type `Quantité > 0` s'en charge automatiquement. Et si *toutes* les lignes finissent exclues, le module n'envoie pas de fichier vide pour autant : rien n'est généré pour cet enregistrement.

Chemin des lignes : imbriquer les lignes dans un conteneur (XML uniquement)

Par défaut, les éléments de ligne répétés sont accrochés directement sous l'élément racine. Certains partenaires les attendent plutôt regroupés dans leur propre conteneur — par exemple une simple balise `<lines>` qui contient toutes les lignes, ou même tout un niveau supplémentaire entre la racine et le document lui-même. Le « **Chemin des lignes** », visible juste sous le Filtre sur les lignes dans la figure ci-dessus, indique au module où les accrocher, sous forme de chemin séparé par des « / ».

💡 Réutiliser le même conteneur qu'un champ d'en-tête

Si le Source Path d'un mapping d'en-tête crée déjà un conteneur (ex: `goodsReceipt/idHeader` , qui imbrique tout un niveau sous une racine `goodsReceipts`), réglez le Chemin des lignes sur `goodsReceipt/lines` — le module réutilise ce même élément `<goodsReceipt>` au lieu d'en créer un second, si bien que les lignes finissent exactement là où elles doivent être : `<goodsReceipts>`
`<goodsReceipt>...<lines>...</lines></goodsReceipt></goodsReceipts>` . Laissez le champ vide pour accrocher les lignes directement sous la racine, comme avant.

💡 Plusieurs commandes dans un seul fichier

Certains partenaires regroupent plusieurs commandes dans un seul fichier EDIFACT envoyé en une fois (un "interchange" contenant plusieurs documents). Le module le détecte automatiquement et crée **une commande distincte par document**, chacune avec ses propres lignes — sans jamais mélanger les informations de deux commandes différentes.

Même logique pour un fichier **CSV** qui contiendrait plusieurs commandes les unes après les autres (une ligne par produit, avec la référence de commande répétée sur chaque ligne) : le module regroupe automatiquement les lignes qui partagent les mêmes informations d'en-tête.

Et en **largeur fixe**, une "ligne de produit" peut elle-même être écrite sur plusieurs lignes physiques du fichier (par exemple trois lignes techniques qui décrivent ensemble un seul article livré). Le tag d'enregistrement marque le début de chaque nouveau groupe — tout ce qui suit, jusqu'au prochain, lui appartient.

Onglet Mappings d'en-tête

C'est ici que se trouve le cœur du paramétrage : chaque ligne du tableau dit "va chercher telle information dans le fichier, et mets-la dans tel champ Odoo".

EDI Échanges Déclencheurs Configuration

Nouveau Profils EDI Commandes partenaire (exemple ORDERS) 3 Échanges 1/3

Nom : Commandes partenaire (exemple ORDERS) Modèle Odoo : Commande client
Format : EDIFACT Canal : AS2 Groupe Distributeur
Direction : Entrant

LIGNES IMBRIQUÉES (OPTIONNEL)

Modèle des lignes ? : Ligne de commande
Champ lignes (one2many) : Lignes de commande (Commande client)
Filtre sur les lignes ? : Faire correspondre tous les enregistrements
Nouvelle règle → 4 enregistrement(s)

Configuration	Mappings d'en-tête	Mappings de lignes	Règles d'action						
Chemin source	Champ Odoo	Chemin de champ	Transformer	Expression	Champ de recherche	Valeur par défaut	Clé upsert	Obligatoire	
BGM.2	Référence client (Commande client)		Aucun				☒	☐	
NAD[BY].2.1	Client (Commande client)		Many2one par référence		ref		☐	☐	

Ajouter une ligne

Mappings d'en-tête — chaque ligne relie un emplacement du fichier à un champ Odoo

- 1 Source Path : l'emplacement exact de la donnée dans le fichier reçu (la syntaxe dépend du format — voir encadré ci-dessous).
- 2 Clé upsert : quand cette case est cochée, ce champ sert à reconnaître si l'enregistrement existe déjà (pour le mettre à jour au lieu d'en créer un doublon).

Colonne	Ce qu'elle veut dire
Champ Odoo	Le champ de destination dans Odoo (ex: Référence client, Client...)
Chemin de champ	Sortant uniquement — une alternative à « Champ Odoo » quand la valeur se trouve en fait sur un enregistrement lié (ex: <code>partner_id.ref</code> pour lire la référence du client). Laisser « Champ Odoo » vide dans ce cas.
Transformer	Une conversion à appliquer à la valeur lue : la transformer en nombre, en date, ou aller chercher un enregistrement existant (client, produit) par une de ses références.
Expression	Uniquement si le Transformer choisi est « Expression Python » : une petite formule de transformation personnalisée (réservée aux cas particuliers, généralement mise en place avec l'aide d'un consultant).

Champ de recherche	Pour les transformers "Many2one" : quel champ utiliser pour retrouver le client/produit existant (ex: son code-barres, sa référence).
Valeur par défaut	Valeur utilisée si l'information est absente du fichier.
Obligatoire	Sortant uniquement — si cette valeur est vide, le fichier n'est pas envoyé du tout : l'échange est marqué en erreur à la place, avec un message clair, pour que l'information manquante n'arrive jamais silencieusement chez le partenaire.

ED I Échanges Déclencheurs Configuration

Nouveau Profils EDI Confirmation de commande (sortant) ⚙

Échanges 3

3 / 3 < >

Nom Confirmation de commande (sortant) Modèle Odoo Commande client

Format XML Canal Webhook Boutique en ligne

Direction Sortant

LIGNES IMBRIQUÉES (OPTIONNEL)

Modèle des lignes ? Ligne de commande

Champ lignes (one2many) Lignes de commande (Commande client)

Filtre sur les lignes ? Faire correspondre tous règles suivantes :

Quantité supérieur à 0,00

Nouvelle règle → 4 enregistrement(s)

Chemin des lignes ? lines

Configuration Mappings d'en-tête Mappings de lignes Règles d'action

Chemin so...	Champ Odoo	Chemin de ch...	Transformer	Expression	Champ de recher...	Valeur par dé...	Clé up...	Obligat...
Ref	Référence client (Commande cli...	Aucun						
Customer	Client (Commande client)	Aucun						
ShopRef	partner_id.ref	Aucun						
Composed...		expression Pyt...		'CMD-%s-%s' % (record.name, record.partner_id.na...				

Ajouter une ligne

Trois colonnes réservées au sortant, en action, sur les mêmes Mappings d'en-tête

- 1 Chemin de champ : lit partner_id.ref sur le client lié, puisque « Champ Odoo » seul ne peut pas atteindre un enregistrement lié.
- 2 Obligatoire : coché ici — si cette valeur revient vide, le fichier n'est pas envoyé, l'échange passe en erreur à la place.
- 3 Transformer réglé sur « Expression Python » : compose une toute nouvelle valeur à partir de deux champs à la fois (voir l'encadré ci-dessous).

💡 Composer une valeur à partir de plusieurs champs (sortant)

Parfois un partenaire attend une valeur unique construite à partir de deux champs Odoo ou plus, collés ensemble — par exemple `CMD-S00002-Norda Retail Group` dans la capture ci-dessus, combinant une référence de commande et un nom de client avec un préfixe fixe. Aucun des transformers ci-dessous ne peut faire ça seul, puisque chaque mapping lit normalement un seul champ.

Expression Python le peut : pour les mappings sortants, `value` (la valeur résolue de ce mapping) et `record` (l'enregistrement entier en train d'être envoyé) sont tous les deux disponibles dans le champ Expression. `'CMD-%s-%s' % (record.name, record.partner_id.name)` produit exactement cette référence composée — « Champ Odoo » et « Chemin de champ » peuvent même rester vides dans ce cas, puisque l'expression construit la valeur entièrement à partir de `record`.

💡 Toutes les options de "Transformer"

Transformer	Effet
Aucun	La valeur est utilisée telle quelle, sans conversion (texte brut).
Date (YYYY-MM-DD)	Interprète la valeur comme une date écrite année-mois-jour (le format que Odoo utilise en interne).
Date (DD/MM/YYYY)	Idem, mais pour une date écrite jour/mois/année — le format européen courant utilisé par de nombreux partenaires.
Nombre décimal	Convertit la valeur en nombre avec décimales (ex: un prix ou une quantité comme 12.5).
Entier	Convertit la valeur en nombre entier (ex: une quantité comme 12).
Booléen (true/1/yes)	Convertit la valeur en case à cocher (Oui/Non), en reconnaissant "true", "1" ou "yes" (sans distinction de casse) comme cochée.
Many2one par nom	Recherche un client/produit existant par son <i>nom</i> (ou le champ choisi dans "Champ de recherche") au lieu d'en créer un nouveau.
Many2one par référence	Même principe, mais recherche par <i>référence</i> (code-barres, code interne...) plutôt que par nom — le choix habituel en EDI, car les noms correspondent rarement exactement entre systèmes.
Expression Python	Une formule personnalisée pour les cas particuliers (voir "Expression" ci-dessus) — réservée aux situations que les autres transformeurs ne peuvent pas gérer.

💡 Comprendre le "Source Path" selon le format

- **XML** : le chemin des balises, ex. `Header/RefCmd` ou `@Ref` pour un attribut.
- **JSON** : la clé, ex. `ref` ou `adresse/ville` pour une valeur imbriquée dans un sous-objet.
- **EDIFACT** : une notation compacte `SEGMENT.élément.composant` — par exemple `BGM.2` veut dire « dans le segment BGM, le 2^e élément ». Quand un segment peut apparaître plusieurs fois avec un rôle différent (ex: l'adresse de l'acheteur ou du fournisseur), on précise un qualifiant entre crochets : `NAD[BY].2.1` signifie « le segment NAD qualifié BY (acheteur), 2^e élément, 1^{er} composant ». C'est la partie la plus technique du module — elle demande de connaître un peu la structure EDIFACT du partenaire, généralement fournie dans son manuel d'intégration.

- **Largeur fixe** : `CODE.position.longueur` — par exemple `HH.2.10` veut dire « sur la ligne qui commence par HH, lire 10 caractères à partir de la position 2 ». La position se compte à partir de 0 (le tout premier caractère de la ligne).
- **CSV** : simplement le nom de la colonne (ex: `reference_client`), ou son numéro si le fichier n'a pas de ligne d'en-tête.

Onglet Mappings de lignes

Si le profil a des « lignes imbriquées » activées, un second tableau identique apparaît, mais pour les champs des lignes de produits (quantité, produit, description...) plutôt que de l'en-tête.

ED I Échanges Déclencheurs Configuration

Nouveau Profils EDI Commandes partenaire (exemple ORDERS) Échanges 3 1/3 < >

Nom Commandes partenaire (exemple ORDERS) Modèle Odoo Commande client

Format EDIFACT Canal AS2 Groupe Distributeur

Direction Entrant

LIGNES IMBRIQUÉES (OPTIONNEL)

Modèle des lignes ? Ligne de commande

Champ lignes (one2many) Lignes de commande (Commande client)

Filtre sur les lignes ? Faire correspondre **tous les enregistrements**

[Nouvelle règle](#)

→ 4 enregistrement(s)

Configuration Mappings d'en-tête **Mappings de lignes** Règles d'action

Chemin source	Champ Odoo	Chemin de champ	Transformer	Expression	Champ de recherche	Valeur par défaut
LIN.3.1	Produit (Ligne de commande)		Many2one par référence		barcode	
QTY.1.2	Quantité (Ligne de commande)		Nombre décimal			
IMD.3.4	Description (Ligne de commande)		Aucun			

[Ajouter une ligne](#)

Mappings de lignes — une ligne EDIFACT (LIN) devient une ligne de commande Odoo

Onglet Règles d'action

Une fois toutes les valeurs récupérées, il faut dire à Odoo quoi en faire : créer un nouvel enregistrement, mettre à jour un existant, ou les deux selon le cas (« upsert », le choix le plus courant).

EDI

Échanges

Déclencheurs

Configuration

Nouveau

Profils EDI

Commandes partenaire (exemple ORDERS) ⚙

Échanges

3

1 / 3

Nom

Commandes partenaire (exemple ORDERS)

Modèle Odoo

Commande client

Format

EDIFACT

Canal

AS2 Groupe Distributeur

Direction

Entrant

LIGNES IMBRIQUÉES (OPTIONNEL)

Modèle des lignes ?

Ligne de commande

Champ lignes (one2many)

Lignes de commande (Commande client)

Filtre sur les lignes ?

Faire correspondre **tous les enregistrements**

Nouvelle règle

→ 4 enregistrement(s)

Configuration

Mappings d'en-tête

Mappings de lignes

Règles d'action

Nom	Type d'action	Nom de la méthode	Condition	Actif	
Créer ou mettre à jour	Créer ou mettre à jour			☒	
Ajouter une ligne					

Règles d'action — que faire une fois les données extraites

Action	Effet
Créer	Crée toujours un nouvel enregistrement, même si un similaire existe déjà.
Mettre à jour	Modifie un enregistrement existant (trouvé grâce à la « Clé upsert »). Ne fait rien si aucun n'est trouvé.
Créer ou mettre à jour	Le choix habituel : met à jour si l'enregistrement existe, le crée sinon.
Appeler une méthode	Déclenche une action Odoo existante sur l'enregistrement (ex: confirmer une commande).
Notifier	Ajoute simplement un message sur la fiche, sans la modifier.

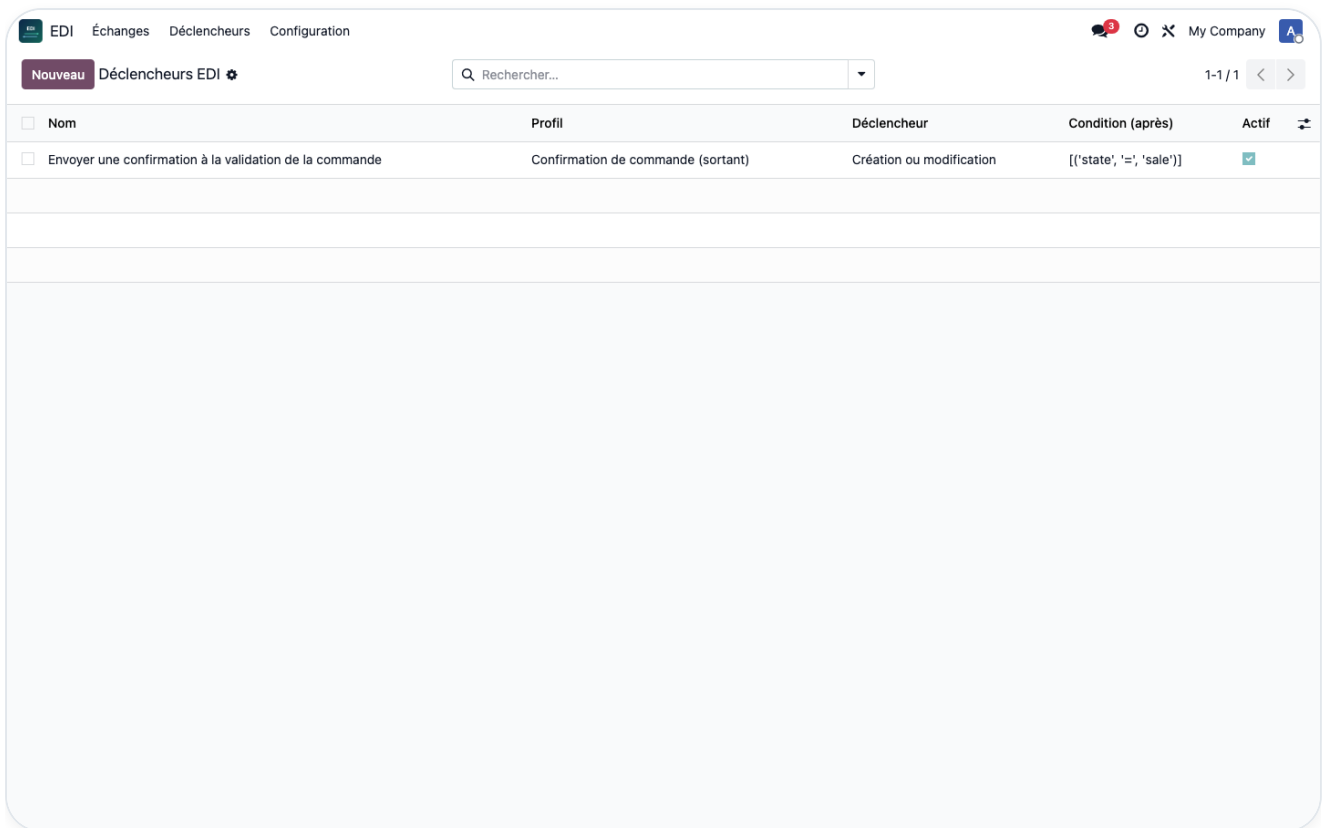
💡 "Condition" : n'exécuter une règle que dans certains cas

Chaque règle d'action peut porter une **Condition** optionnelle — une petite formule qui doit être vraie pour que cette règle s'exécute, vérifiée sur les valeurs qui viennent d'être extraites du fichier (ex: `values.get('state') == 'done'` n'exécute la règle que si le statut de la commande dans le fichier est « done »).

C'est ce qui permet à plusieurs règles de coexister sur le même profil pour des cas différents — par exemple « Appeler une méthode » (confirmer la commande) uniquement quand le statut est « done », et « Notifier » comme filet de sécurité sinon. Laisser vide pour une règle qui doit toujours s'exécuter.

6. Les déclencheurs : automatiser l'envoi

Jusqu'ici, tous les exemples concernaient des fichiers **reçus**. Mais le module peut aussi **envoyer** automatiquement un fichier dès qu'un enregistrement Odoo change — sans bouton à cliquer, sans action manuelle. C'est le rôle des **déclencheurs**.



Déclencheurs EDI				
Nom	Profil	Déclencheur	Condition (après)	Actif
☐ Envoyer une confirmation à la validation de la commande	Confirmation de commande (sortant)	Création ou modification	[('state', '=', 'sale')]	☒

Liste des déclencheurs configurés

Un déclencheur relie trois choses : un **profil sortant** (celui qui sait construire le fichier), un **événement** (création, modification, ou suppression d'un enregistrement), et des **conditions** optionnelles pour ne se déclencher que dans certains cas précis.

EDI

Échanges

Déclencheurs

Configuration

Nouveau

Déclencheurs EDI

Envoyer une confirmation à la validation de la commande

1/1

Nom

Envoyer une confirmation à la validation de la commande

Déclencheur

Création ou modification

Profil

Confirmation de commande (sortant)

Automatisation

[EDI] Envoyer une confirmation à la validation de la commande

FILTRES

Condition (après) ?

Faire correspondre tous ▾ règles suivantes :

Statut

est égal à

Commande client

Nouvelle règle

→ 1 enregistrement(s)

Condition (avant modification) ?

Faire correspondre tous ▾ règles suivantes :

Statut

est différent de

Commande client

Nouvelle règle

→ 1 enregistrement(s)

AIDE

Exemple SO confirmée : Trigger = "Création ou modification", Condition après = [{"state", "=", "sale"}], Condition avant = [{"state", "!=", "sale"}]

Un déclencheur — envoyer une confirmation dès qu'une commande passe au statut « Commande client »

Champ	Ce qu'il veut dire
Profil	Le profil sortant à utiliser pour construire le fichier envoyé (seuls les profils entrant/sortant ou sortant apparaissent ici).
Déclencheur	Quand vérifier les conditions : à la création, à la création ou modification, ou à la suppression de l'enregistrement.
Condition (après)	Le fichier n'est envoyé que si l'enregistrement, une fois l'action faite, satisfait ce filtre (ex: la commande est bien confirmée).
Condition (avant modification)	Optionnelle — permet de ne déclencher que sur un vrai <i>changement d'état</i> (ex: la commande n'était PAS encore confirmée avant), plutôt qu'à chaque sauvegarde d'un enregistrement déjà confirmé.

📌 Un filtre visuel, pas une formule à taper

Les deux conditions utilisent le même constructeur de filtre visuel que le « Filtre sur les lignes » vu plus haut pour les profils — on choisit un champ, un opérateur, une valeur, et le domaine sous-jacent est construit automatiquement. Les exemples ci-dessous sont montrés sous leur forme brute uniquement pour expliquer ce qui est réellement vérifié.

💡 L'exemple le plus courant : confirmer une commande

Envoyer automatiquement une confirmation dès qu'une commande passe au statut « Commande client » :
 Déclencheur = « Création ou modification », Condition (après) = [{"state", "=", "sale"}], Condition

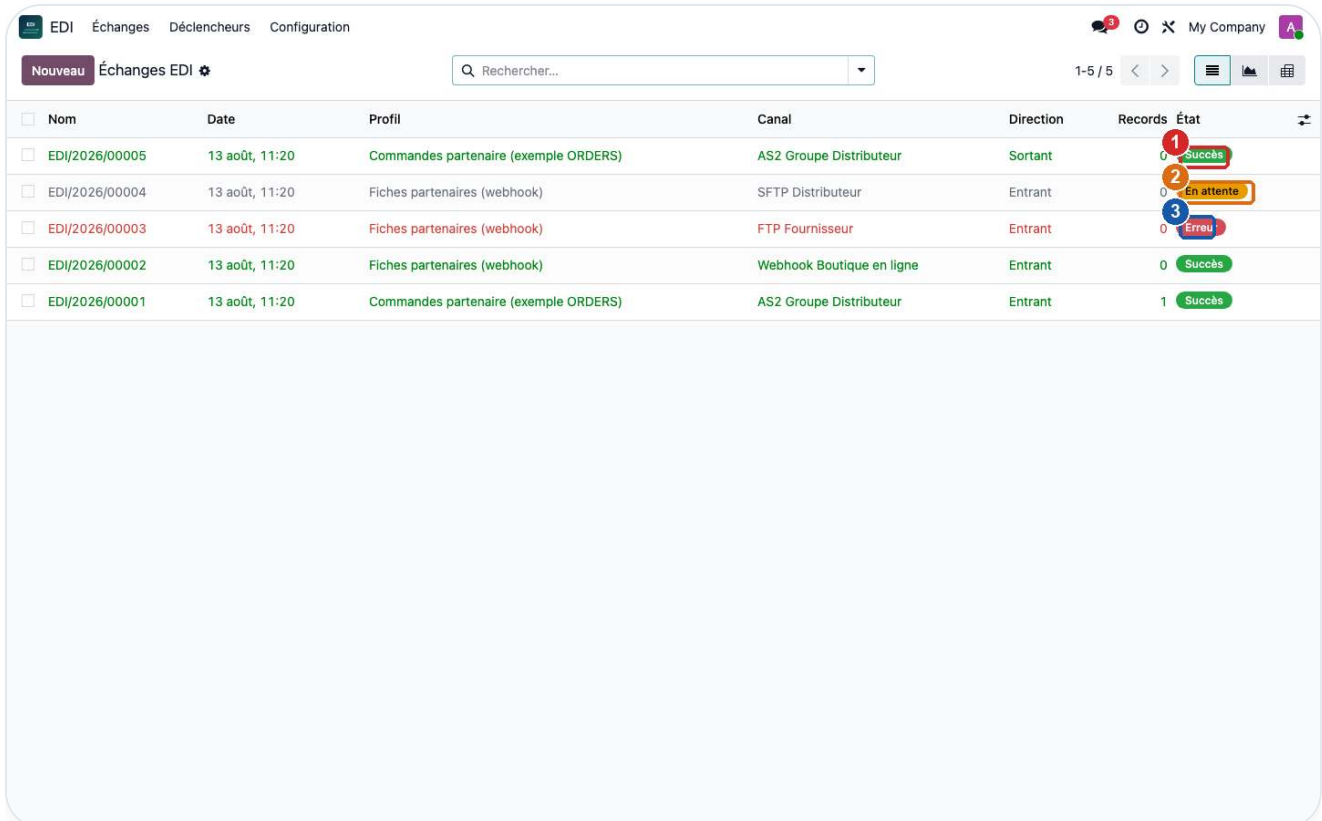
(avant modification) = `[('state', '!=', 'sale')]` — ce dernier filtre évite de renvoyer le fichier à chaque modification ultérieure d'une commande déjà confirmée.

Sous le capot

Chaque déclencheur crée automatiquement une automatisation Odoo standard (visible dans Réglages → Technique) — pas de mécanisme parallèle à maintenir, le module s'appuie sur l'automatisation native d'Odoo.

7. Suivre les échanges au quotidien

C'est l'écran que vous ouvrirez le plus souvent : la liste de tout ce qui est passé par le module, avec un code couleur clair.



The screenshot shows the 'Échanges EDI' interface. At the top, there's a navigation bar with 'EDI', 'Échanges', 'Déclencheurs', and 'Configuration'. Below it, a 'Nouveau' button and a search bar labeled 'Rechercher...'. The table has columns: 'Nom', 'Date', 'Profil', 'Canal', 'Direction', 'Records', and 'État'. The 'État' column contains status indicators: 'Succès' (green), 'En attente' (orange), and 'Erreur' (red). The table lists five records, all dated '13 août, 11:20'. The first record is 'Succès', the second is 'En attente', the third is 'Erreur', the fourth is 'Succès', and the fifth is 'Succès'.

Nom	Date	Profil	Canal	Direction	Records	État
EDI/2026/00005	13 août, 11:20	Commandes partenaire (exemple ORDERS)	AS2 Groupe Distributeur	Sortant	0	Succès
EDI/2026/00004	13 août, 11:20	Fiches partenaires (webhook)	SFTP Distributeur	Entrant	0	En attente
EDI/2026/00003	13 août, 11:20	Fiches partenaires (webhook)	FTP Fournisseur	Entrant	0	Erreur
EDI/2026/00002	13 août, 11:20	Fiches partenaires (webhook)	Webhook Boutique en ligne	Entrant	0	Succès
EDI/2026/00001	13 août, 11:20	Commandes partenaire (exemple ORDERS)	AS2 Groupe Distributeur	Entrant	1	Succès

Liste des échanges avec leur statut

- 1 Succès (vert) : tout s'est bien passé, l'enregistrement Odoo a été créé ou mis à jour.
- 2 En attente (orange) : le fichier est arrivé mais n'a pas encore été traité.
- 3 Erreur (rouge) : quelque chose a empêché le traitement — voir plus bas comment lire le détail.

Filtrer et regrouper

Le menu de recherche propose des filtres rapides prêts à l'emploi.

EDIÉchangesDéclencheursConfiguration

NouveauÉchanges EDI

Rechercher...

1-5 / 5

Nom

Date

EDI/2026/00005

13 août,

EDI/2026/00004

13 août,

EDI/2026/00003

13 août,

EDI/2026/00002

13 août,

EDI/2026/00001

13 août,

Filtres

Erreurs

Succès

En attente

Aujourd'hui

Entrant

Sortant

Filtre personnalisé...

Regrouper par

Profil

Canal

État

Direction

Date

Groupe personnalisé

Favoris

Enregistrer la recherch...

Direction

Records

État

Sortant

0

Succès

Entrant

0

En attente

Entrant

0

Erreur

Entrant

0

Succès

Entrant

1

Succès

Filtres rapides : erreurs, aujourd'hui, entrant/sortant, regroupement par profil ou canal...

Le détail d'un échange

En ouvrant un échange, vous voyez son contenu brut, sa progression, et — une fois traité avec succès — un accès direct à ce qui a été créé.

EDI

Échanges

Déclencheurs

Configuration

Nouveau

Échanges EDI

EDI/2026/00001

Records

1

5 / 5

En attente

Succès

Erreur

Nom

EDI/2026/00001

Direction

Entrant

Profil

Commandes partenaire (exemple ORDERS)

Date

13 août, 11:20

Canal

AS2 Groupe Distributeur

Contenu

Records traités

```

1  UNB:+.? '
2  UNB+UNOC:3+5410000000001:14+5410000000000:14+260717:1140+1361++BELU_V2'
3  UNH+1361+ORDERS:D:018:UN:EAN010'
4  BGM+220+PO-2026-00042+9'
5  DTM+137:20260717:102'
6  DTM+2:20260729:102'
7  NAD+BY+5410000000001::9'
8  NAD+SU+5410000000001::9'
9  LIN+1+54100000000101:SRV'
10 IMD+F++::Solstice Ice Tea 33cl'
11 QTY+21:162'
12 LIN+2+54100000000102:SRV'
13 IMD+F++::Nordic Sesame Bar 20g'
14 QTY+21:30'
15 UNS+S'
16 UNT+13+1361'
17 UNZ+1+1361'

```

Détail d'un échange réussi

- 1 La barre de statut montre où en est le traitement.
- 2 Le bouton « Records » ouvre directement l'enregistrement Odoo créé ou modifié par cet échange.

EDI

Échanges

Déclencheurs

Configuration

Nouveau

Échanges EDI

EDI/2026/00001

Records

1

5 / 5

En attente

Succès

Erreur

Nom

EDI/2026/00001

Direction

Entrant

Profil

Commandes partenaire (exemple ORDERS)

Date

13 août, 11:20

Canal

AS2 Groupe Distributeur

Contenu

Records traités

```

[{"model": "sale.order", "id": 1}]

```

💡 « Retraiter » vs « Réessayer » : quelle différence ?

Retraiter ne concerne que l'entrant — il relance simplement l'analyse du fichier déjà reçu, tant qu'il n'est pas déjà réussi. **Réessayer** ne concerne que le sortant — il n'apparaît que quand l'échange est en erreur, et régénère entièrement le contenu depuis l'enregistrement source avant de le renvoyer, le bouton à utiliser une fois l'information manquante corrigée.

📌 « Renvoyer » : renvoyer exactement le même fichier

Visible sur un échange sortant **réussi**, « Renvoyer » couvre un cas différent de Réessayer : rien n'a échoué de votre côté, mais le partenaire redemande le fichier (perdu chez lui, souci de traitement...). Il renvoie exactement le même contenu, bit à bit, sans le régénérer — le partenaire reçoit donc précisément ce qui avait été accusé réception la première fois. Une popup de confirmation apparaît avant l'envoi, et un nouvel échange est créé pour l'occasion, préservant le succès de l'original dans l'historique.

Comprendre une erreur

Quand le statut est « Erreur », un onglet supplémentaire apparaît avec le message d'explication.

The screenshot shows the Odoo EDI interface. At the top, there are tabs for 'EDI', 'Échanges', 'Déclencheurs', and 'Configuration'. Below these, there's a 'Nouveau' button and a dropdown menu showing 'Échanges EDI' and 'EDI/2026/00003'. On the right, there's a user profile icon labeled 'My Company' and a notification bell icon. The main content area has a 'Retraiter' button on the left and three status tabs on the right: 'En attente', 'Succès', and 'Erreur' (which is selected). Below the tabs, there's a table with columns 'Nom', 'Direction', 'Entrant', 'Profil', 'Date', and 'Canal'. The table contains one row with the following data: Nom: EDI/2026/00003, Direction: Entrant, Profil: Fiches partenaires (webhook), Date: 14 août, 14:57, Canal: FTP Fournisseur. Below the table, there are two tabs: 'Contenu' and 'Erreur' (which is selected). The 'Erreur' tab displays the message: 'Champ 'name' manquant, contrainte de base de données violée'.

Détail d'un échange en erreur, avec le message explicatif

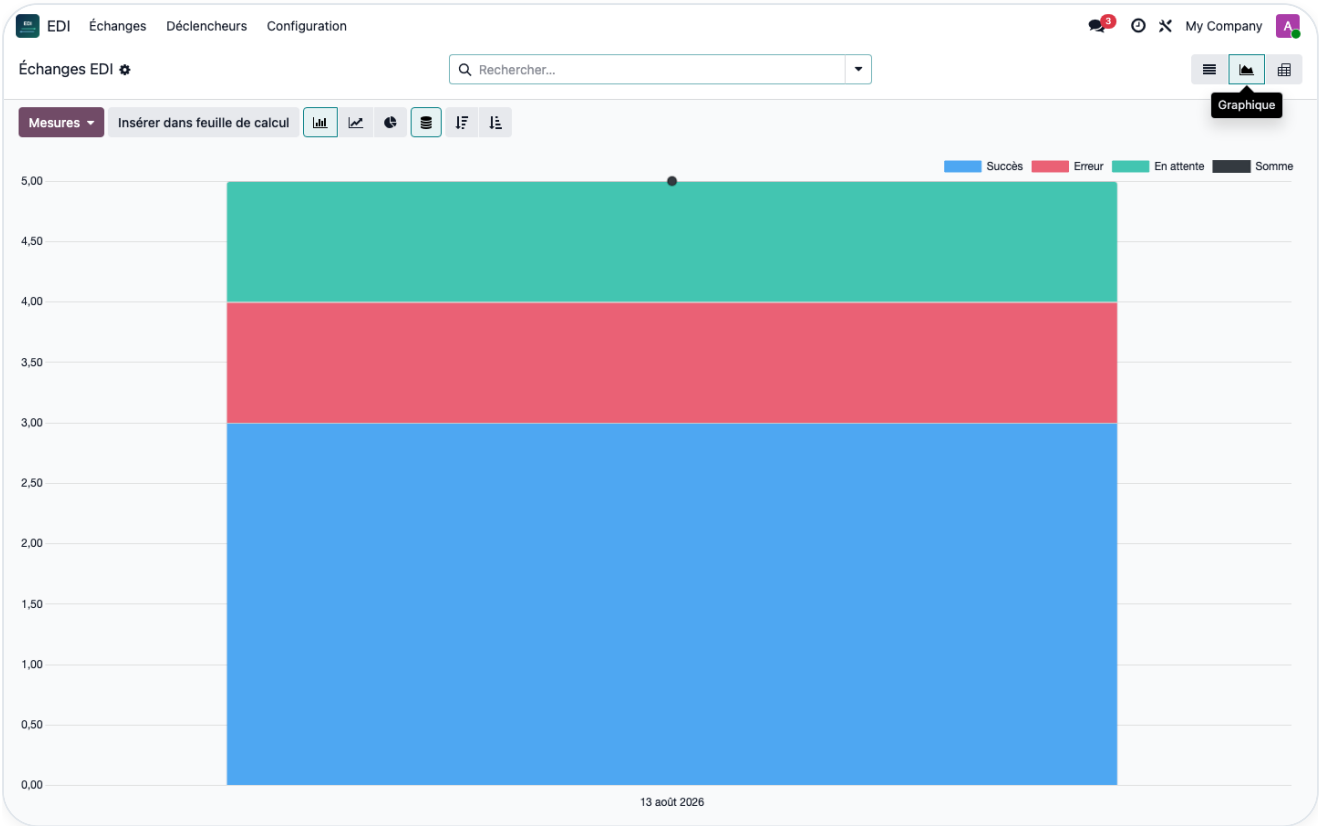
⚠️ Que faire en cas d'erreur ?

Le message d'erreur donne presque toujours la cause exacte (ex: "champ obligatoire manquant", "aucun partenaire trouvé"). Une fois la cause corrigée (dans Odoo ou en redemandant un fichier correct au partenaire),

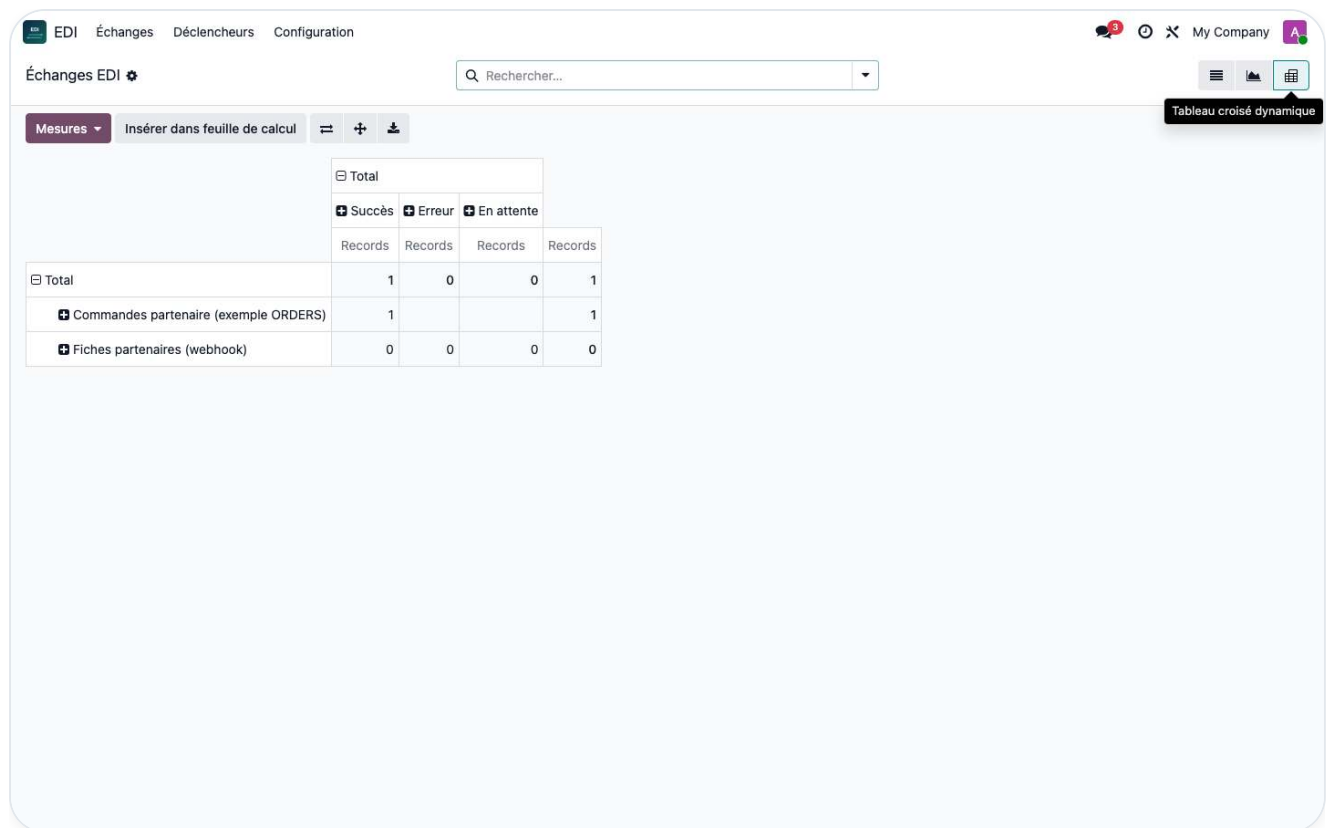
relancez : « **Retraiter** » pour un fichier entrant (ici en exemple) — ou « **Réessayer** » pour un fichier sortant, qui régénère le fichier avant de le renvoyer.

Vue graphique et tableau croisé

Pour surveiller les volumes dans le temps (par exemple, repérer un partenaire qui génère beaucoup d'erreurs), deux vues d'analyse sont disponibles depuis les mêmes icônes en haut à droite de la liste.



Vue graphique : volumes d'échanges par jour et par statut



Vue tableau croisé : répartition par profil et par statut

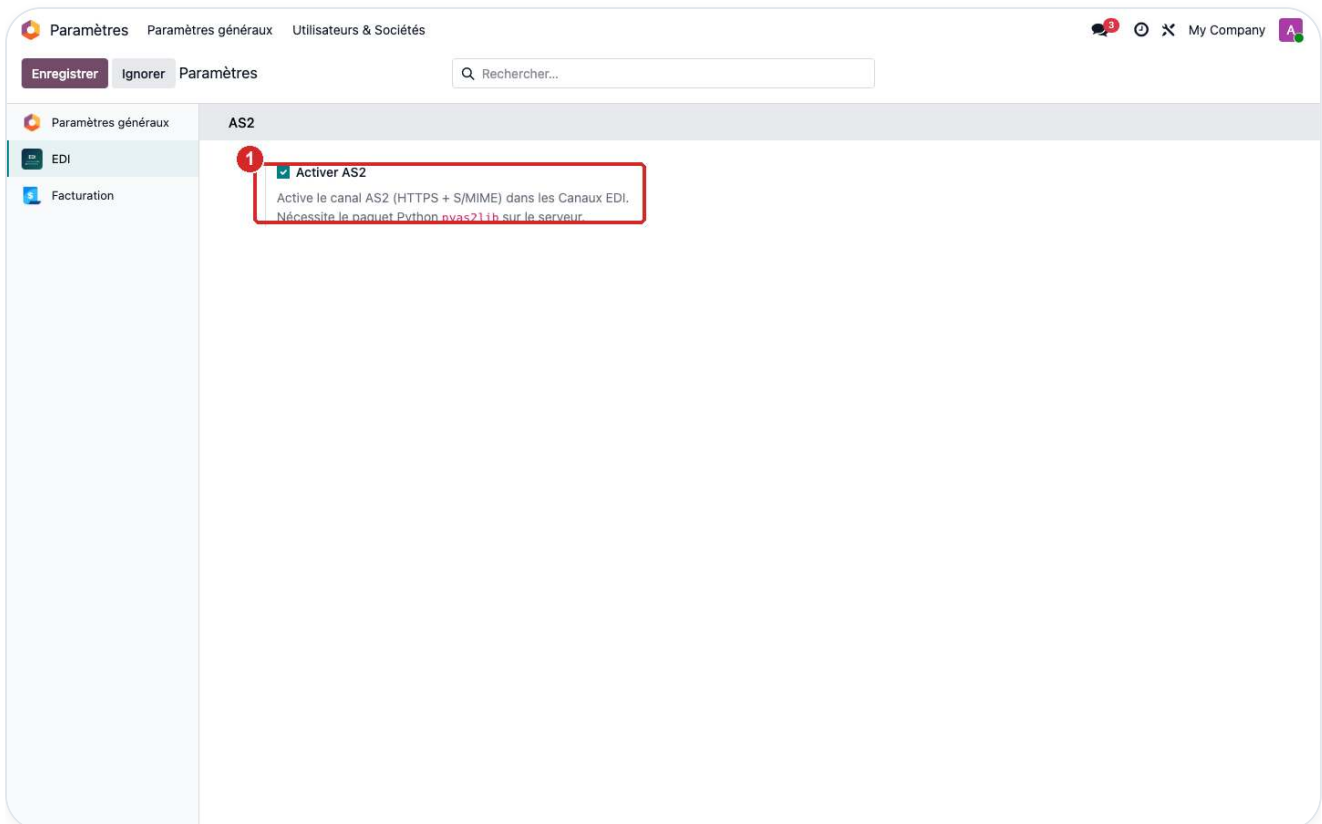
8. Le protocole AS2 en détail

AS2 est le protocole que la plupart des grands distributeurs et industriels imposent à leurs fournisseurs. Il ajoute trois garanties que les autres canaux n'ont pas nativement :

- **Signature** : le fichier est signé électroniquement, ce qui prouve qu'il vient bien de vous (et pas d'un tiers).
- **Chiffrement** : le contenu est illisible pour quiconque l'interceptait en transit.
- **Accusé de réception (MDN)** : le partenaire renvoie une preuve cryptographique qu'il a bien reçu et déchiffré le fichier — une preuve légale en cas de litige ("je n'ai jamais reçu votre commande").

⚠ Interrupteur général obligatoire

Avant de pouvoir utiliser un canal AS2, il faut activer la fonctionnalité dans les réglages généraux d'Odoo. Sans ça, tous les canaux AS2 restent bloqués, même correctement configurés.



Réglages → EDI : l'interrupteur qui active AS2 dans tout le module

1 Cette case doit être cochée et enregistrée pour que les canaux AS2 fonctionnent.

Les certificats

AS2 repose sur des **certificats électroniques** — un peu comme une carte d'identité numérique. Il en faut trois par canal :

Certificat	Rôle
Notre certificat + notre clé privée	Servent à signer nos messages et déchiffrer ce que le partenaire nous envoie.
Certificat du partenaire	Sert à vérifier sa signature et chiffrer ce qu'on lui envoie.
Certificat CA (optionnel)	Si le partenaire utilise un certificat signé par une autorité de certification reconnue plutôt qu'auto-signé, ce certificat permet de vérifier toute la chaîne de confiance.

Ces fichiers sont en général fournis par le service informatique du partenaire, ou par le vôtre s'il gère déjà ce type de certificat pour d'autres échanges.

Mode synchrone ou asynchrone

L'accusé de réception (MDN) peut arriver immédiatement dans la même connexion (**synchrone**, le plus simple et le plus courant), ou être renvoyé séparément un peu plus tard par le partenaire (**asynchrone**) — dans ce cas, le module sait automatiquement associer cet accusé différé au bon échange dès qu'il arrive.

Sur quoi se base cette association ?

Pas sur une donnée métier comme la référence de commande — sur un **identifiant technique du protocole AS2** (le "Message-ID"), généré automatiquement à l'envoi et indépendant du contenu du fichier. Quand le partenaire renvoie son accusé plus tard, il renvoie ce même identifiant ; le module retrouve l'échange correspondant grâce à lui, même si le fichier envoyé est chiffré et illisible en dehors d'Odoo.

9. Exemple complet de bout en bout

Pour rendre tout ça concret, voici un exemple représentatif : un distributeur envoie une commande au format EDIFACT, contenant deux produits.

Le fichier reçu

Voici, dans le détail d'un échange, le fichier brut tel qu'il arrive — illisible à l'œil nu, mais parfaitement structuré pour le module :

The screenshot shows the Odoo EDI interface. At the top, there's a navigation bar with 'EDI', 'Échanges', 'Déclencheurs', and 'Configuration'. Below this, a 'Nouveau' button and a 'Records' button (highlighted with a red circle '2') are visible. The main area displays details for an EDI exchange: 'Nom: EDI/2026/00001', 'Profil: Commandes partenaire (exemple ORDERS)', and 'Canal: AS2 Groupe Distributeur'. It also shows 'Direction: Entrant' and 'Date: 13 août, 11:20'. A status bar at the top right indicates 'En attente' (highlighted with a red circle '1'), 'Succès', and 'Erreur'. Below this, there are tabs for 'Contenu' and 'Records traités'. The 'Contenu' tab is active, showing a list of 17 lines of EDIFACT data, including headers like 'UNA', 'UNB', 'UNH', 'BGM', 'DTM', 'NAD', 'LIN', 'IMD', 'QTY', 'UNS', and 'UNT'.

Le fichier EDIFACT brut reçu du partenaire

Le résultat dans Odoo

Grâce au profil configuré en section 5, ce fichier devient automatiquement **une commande de vente avec ses deux lignes**, le client déjà identifié :

Paramètres Paramètres généraux Utilisateurs & Sociétés

Nouveau

Devis

S00001

1/1

Envoyer

Imprimer

Confirmer

Aperçu

Annuler

Devis

Envoyé

Commande client

Envoyer message

Note

Activité

S00001

1

Clien: Norda Retail Group

Anvers

Expiration ?

12 sept.

Conditions de paiement

Immédiat

Lignes de commande

Autres informations

Produit	Quantité	Prix unitaire	Taxes	Montant
2 Solstice Ice Tea 33cl	162,00	1,20	15%	\$ 194,40
Nordic Sesame Bar 20g	30,00	0,80	15%	\$ 24,00

Ajouter un produit Ajouter une section Ajouter une note Catalogue

Conditions générales...

Montant hors taxes:

\$ 218,40

Tax 15%:

\$ 32,76

Total:

\$ 251,16

Aujourd'hui

A

Administrator 11:20

Sales Order créé

La commande de vente créée automatiquement, avec ses lignes correspondant exactement au fichier reçu

- 1 Le client a été retrouvé automatiquement grâce à son identifiant GLN présent dans le fichier.
- 2 Les deux lignes de produits, avec les bonnes quantités (162 et 30), viennent directement des segments LIN et QTY du fichier EDIFACT.

💡 Aucune ressaisie

Personne n'a tapé le nom du client, ni "162", ni le nom des produits. Tout vient du fichier — le rôle du profil est justement de dire, une fois pour toutes, où trouver chaque information.

10. Qui a le droit de faire quoi

Le module définit deux niveaux d'accès, pour que les équipes opérationnelles puissent surveiller les échanges sans avoir les droits de modifier la configuration technique.

Groupe	Peut faire	Ne peut pas faire
Utilisateur EDI	Voir les canaux et profils, consulter et réessayer les échanges	Modifier un canal, un profil, ou supprimer un échange
Gestionnaire EDI	Tout, y compris créer/modifier canaux et profils	—

Ces groupes se gèrent comme n'importe quel droit d'accès Odoo, depuis la fiche d'un utilisateur.

11. Questions fréquentes et dépannage

Un échange est resté "En attente", que faire ?

Pour un canal FTP/SFTP, cela signifie généralement que le traitement automatique périodique n'a pas encore tourné — c'est normal d'attendre quelques minutes. Vous pouvez aussi le déclencher manuellement avec le bouton « Réessayer » sur l'échange, ou « Déclencher le polling » sur le canal.

J'ai activé AS2 mais rien ne se passe

Vérifiez dans l'ordre : (1) l'interrupteur dans Réglages → EDI est bien coché *et enregistré*, (2) les identifiants AS2 des deux côtés sont bien remplis, (3) les certificats sont bien chargés, (4) l'URL distante du partenaire est correcte.

Comment savoir si mon canal fonctionne avant le premier vrai envoi ?

Utilisez le bouton « Tester la connexion » sur la fiche du canal (disponible pour Webhook, FTP et SFTP). Pour AS2, le premier test se fait généralement avec un fichier réel envoyé par le partenaire en environnement de test.

Un même partenaire envoie parfois des commandes, parfois des factures — comment le module fait la différence ?

C'est le rôle du **tag racine / type de message** configuré sur chaque profil (section 5). Si plusieurs profils sont rattachés au même canal, le module regarde le contenu du fichier pour choisir automatiquement le bon profil.

Le partenaire a changé le format de ses fichiers, dois-je refaire toute la configuration ?

Non — il suffit d'ajuster les lignes de mapping concernées (le "Source Path") dans le profil existant. Aucun code à modifier.

Puis-je voir exactement ce qui a été envoyé à un partenaire ?

Oui, chaque échange sortant garde le contenu exact du fichier envoyé, consultable à tout moment dans son détail (section 7).

Comment savoir quel format choisir pour un nouveau partenaire ?

Ce n'est pas un choix — c'est le format déjà utilisé par le partenaire qui impose le format du profil. Ouvrez un exemple de fichier réel : des balises `<...>` → XML, des accolades `{...}` → JSON, des blocs `BGM+...'` → EDIFACT, des colonnes séparées par une virgule ou un point-virgule → CSV, et si rien de tout ça mais que chaque ligne a toujours exactement la même longueur → largeur fixe.

12. Glossaire complet

AS2

Protocole de transport EDI sécurisé (signature, chiffrement, accusé de réception), largement imposé dans la grande distribution.

Canal (edi.channel)

La configuration du moyen de transport d'un fichier EDI (Webhook, FTP, SFTP, AS2, Manuel).

Certificat CA

Certificat d'une autorité de certification reconnue, utilisé pour valider qu'un certificat partenaire est authentique et pas seulement auto-signé.

Clé upsert

Champ utilisé pour reconnaître si un enregistrement existe déjà, afin de le mettre à jour plutôt que d'en créer un doublon.

CSV

Format de fichier tabulaire où les valeurs d'une ligne sont séparées par un caractère (virgule, point-virgule...), avec généralement une première ligne d'en-têtes de colonnes.

EDI

Échange de Données Informatisé — l'envoi automatisé de documents commerciaux entre systèmes informatiques, sans ressaisie manuelle.

EDIFACT

Norme internationale historique d'échange EDI, organisée en segments (ex: BGM, NAD, LIN), très utilisée dans le commerce et l'industrie.

Échange (edi.exchange)

L'enregistrement d'un fichier reçu ou envoyé, avec son contenu, son statut et son historique.

GLN

Global Location Number — un identifiant international unique attribué à chaque entreprise ou site, utilisé pour identifier automatiquement les partenaires dans les fichiers EDI.

JSON

Format de données textuel structuré en clés/valeurs, très utilisé par les applications web modernes.

Largeur fixe

Format de fichier texte sans séparateur, où chaque information occupe toujours les mêmes colonnes ; chaque ligne commence en général par un petit code indiquant son type. Format hérité des anciens systèmes "mainframe", encore utilisé par certains distributeurs.

Mapping (edi.field.mapping)

Une ligne de correspondance entre un emplacement dans le fichier source et un champ Odoo.

MDN

Message Disposition Notification — l'accusé de réception cryptographique renvoyé par le partenaire dans un échange AS2.

Profil (edi.profile)

La configuration qui décrit comment lire ou écrire un type de document dans un format donné, et vers quel modèle Odoo.

Qualifiant (EDIFACT)

Un code qui précise le rôle d'un segment répété plusieurs fois dans un même fichier (ex: distinguer l'adresse de l'acheteur de celle du fournisseur, toutes deux dans des segments NAD).

Segment (EDIFACT)

Un bloc d'information dans un fichier EDIFACT, identifié par un code à 3 lettres (BGM, NAD, LIN, QTY...).

Transformer

La conversion appliquée à une valeur lue avant de l'enregistrer dans Odoo (nombre, date, recherche d'un enregistrement existant...).

Webhook

Une adresse web à laquelle un système externe peut envoyer automatiquement des données, ou qu'Odoo peut appeler pour envoyer une donnée à l'extérieur.

XML

Format de données textuel structuré en balises imbriquées, très répandu dans les échanges inter-entreprises.



Une question ? Un problème ?

Contactez l'équipe qui a développé ce module — réponse en moins de 8 heures ouvrées.

Email : cma@nalios.be

Nalios — Odoo Gold Partner, Belgique — nalios.be