

Documentation WeavyToSage



4, rue Samuel Beckett

L-4371 Belvaux

(+352) 26 31 32 511 25

<https://weavytosage.com>

<https://ppi-group.lu>

Table des matières

A.	Présentation du connecteur Weavy To Sage	2
1.	Introduction	2
2.	Contenu du Package de déploiement WeavyToSage	2
3.	Architecture de la solution	2
4.	Les flux traités	3
5.	Les cas non traités	Erreur ! Signet non défini.
B.	Avant de commencer l'implémentation	6
1.	Prérequis	6
I.	Installer et configurer l'API Webservice 100	6
II.	Posséder un code projet Weavy Actif	6
III.	Audit préalable des données utilisées dans Sage	6
C.	Déploiement du connecteur Weavy To Sage	8
1.	WeavyToSage	8
I.	Service	10
II.	Environnements	11
III.	Task	12
IV.	Service Log	24
2.	Studio	25
I.	Variables	25
II.	Dictionnaire de données	25
III.	SrvExport	26
IV.	Designer	26
V.	Backend	27
3.	Control Center	62
I.	Téléchargement / Installation	62
II.	Paramétrage	64
III.	Scénario	65
IV.	Module	66
V.	Lancement du Control Center	79

A. Présentation du connecteur Weavy To Sage

1. Introduction

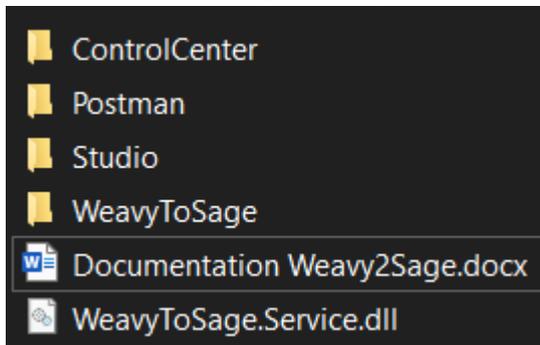
Le présent document a pour but de proposer un guide d'implémentation et d'intégration du connecteur « Plug & Play » WeavyToSage.

WeavyToSage permet de coupler la solution CRM mobile Divalto Weavy à la solution ERP Sage 100 cloud Gestion Commerciale.

Elle donne de plus les clés à l'intégrateur pour réaliser un déploiement rapide de Divalto Weavy et couvre tous les outils de la solution sur ses aspects les plus techniques.

En suivant ce guide pas à pas, le déploiement de la solution Weavy standard avec une Gestion Commerciale Sage 100 cloud standard peut-être réalisée en au plus une journée.

2. Contenu du Package de déploiement WeavyToSage



Dossier « ControlCenter » : Kit de paramétrage complet Control Center prêt à l'emploi (Config et Requêtes Sage)

Dossier « Postman » : Package de test JSON Postman pour tests unitaires de l'API Weavy (voir [Postman](#))

Dossier « Studio » : Kit de paramétrage pour Weavy Studio prêt à l'emploi (Interfaces backend et Scripts)

Dossier « WeavyToSage » : setup du connecteur WeavyToSage

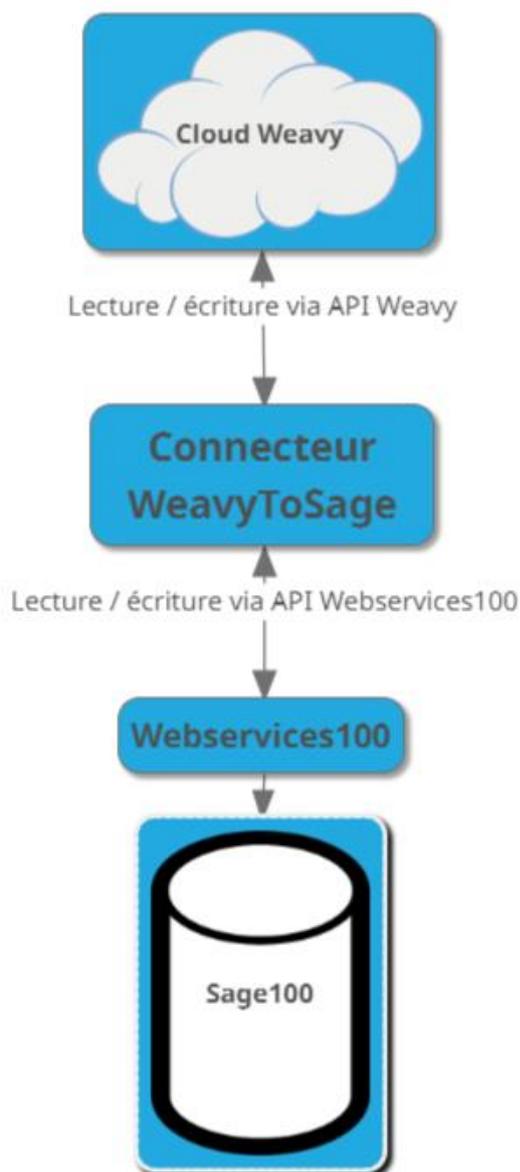
« WeavyToSage.Service.dll » : Patch à copier après l'installation de WeavyToSage ().

3. Architecture de la solution

WeavyToSage est une application basée sur des API webservice.

API Webservices 100 (<http://www.webservices100.com>).

API Webservices Divalto Weavy



Grace à l'API Webservices 100, WeavyToSage vous permet d'avoir une configuration multi-environnement et multi-versions (Sage), ce qui vous permet d'exécuter plusieurs tâches de synchronisation qui ciblent des environnements et donc des bases différentes.

De plus, les montées de version Sage 100cloud sont simplifiées et vos clients pourront bénéficier de chaque mise à jour Sage 100 sans revoir nécessairement vos développements spécifiques.

4. Les flux traités

FLUX	Sage -> Weavy (ControlCenter)	Weavy -> Sage (WeavyToSage)
Clients	X	X
Contacts	X	X
Adresses	X	X
Documents de ventes (les PDF des documents ne sont pas synchronisés)		X (Devis, commande, Bon de livraison)

Interventions (inclus Photos et Fiche Intervention PDF selon paramétrage)		X (Préparation de livraison, Bon de livraison)
Types génériques	X	
Devises	X	
Gammes	X	
Enumérés de gammes	X	
Articles simples	X	
Articles à gammes	X	
Articles à conditionnement de vente	X	
Articles facturés au poids net	X	
Articles sérialisés (Lot ou Série)	X	
Familles produit	X	
Dépôts	X	
Stock article	X	
Commerciaux	X	
Lien Commerciaux / Clients	X	
Encours	X	
Catégories tarifaire	X	
Taxes	X	
Conditions commerciales	X	
Historique entêtes *	X (Préparation de livraison, bon de livraison, facture, facture comptabilisée)	
Historique lignes *	X	
Statistiques **	X (3 dernières années)	

* : choix des pièces à reprendre dans les constantes du ControlCenter.

** : choix de la période à reprendre dans les constantes du ControlCenter.

5. Les cas non traités

Éléments importants non gérés par le connecteur, à identifier en avant-vente / avant-projet.

Certaines options d'articles Sage impactant la méthode de facturation ne sont pas supportées :

- ⚠ Nomenclatures Commerciales Composantes ou Composées de Sage (équivalent aux Packs dans Weavy)
- ⚠ Les options d'articles Sage suivantes (onglet Paramètres de la fiche Article Sage) ne sont pas supportées

- Ne pas soumettre à l'escompte
- Vendre au débit
- Gérer en contremarque
- Facturer au forfait

The screenshot shows a software configuration window with a top navigation bar containing 'Fonctions', 'Nomenclature', 'Interroger', and 'Traçabilité'. Below this is a sub-menu with 'Identification', 'Descriptif', 'Champs Libres', and 'Paramètres'. The main content area is titled 'Options de traitement' and is divided into three sections: 'Facturation', 'Impression', and 'Modèle d'enregistrement'. The 'Facturation' section contains six checkboxes: 'Ne pas soumettre à l'escompte', 'Publier sur le site marchand' (checked), 'Mettre en sommeil', 'Facturer / Poids net', 'Vendre au débit', and 'Gérer en contremarque'. The 'Impression' section contains three checkboxes: 'Ne pas imprimer', 'Facturer au forfait', and 'Exclure des statistiques'. The 'Modèle d'enregistrement' section has two input fields labeled 'Intitulé' and 'Domaine'.

B. Avant de commencer l'implémentation

1. Prérequis

Si vous avez téléchargé le « package_install.zip », il faut penser à débloquer le fichier zip avant de l'utiliser, sinon vous rencontrerez des erreurs liées aux dernières options de sécurité de Windows qu'il convient de libérer.

Vérifiez que vous avez ouvert cette documentation depuis le dossier package_install si vous voulez utiliser les liens dynamiques vers les fichiers du package (scripts SQL, etc.).

I. Installer et configurer l'API Webservice 100

Avoir un Webservices100 avec une licence active.

(Voir <http://www.webservices100.com>) :

Télécharger et Installer la dernière version de l'API Webservices 100

Demander une licence à support@ppi-group.lu.

Activer la licence dans le module d'administration des Webservices 100.

Paramétrer le ou les environnements Webservices (Prod + TEST) correspondants aux bases Sage à connecter

II. Posséder un code projet Weavy Actif

Avoir au minimum un code projet Weavy installé et activé par Divalto.

Idéalement prévoir la mise en place de l'environnement de Développement en plus de l'environnement de Production et faire un déploiement de TEST avant de tout mettre en place en Production.

III. Audit préalable des données utilisées dans Sage

Identifier les flux de remontée de données de Sage 100 cloud vers Weavy et vérifier qu'elles sont correctement paramétrées / remplies :

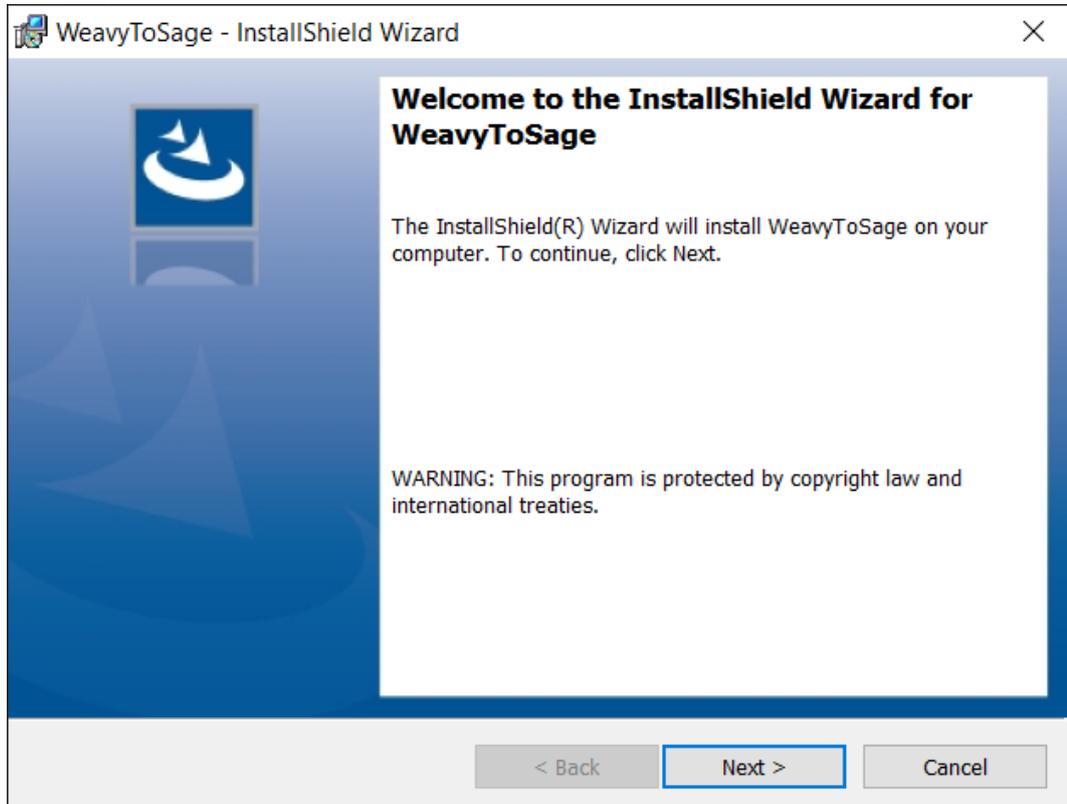
- Données « génériques » (Paramètres société).
 - Unités de Vente : UNIT (P_UNITE)
 - Information libre « Réseau » : RESEAU (F_ENUMLIBRECPTA)
 - Codes Taxes Produits : TAX_PRODUCT (F_TAXE)
 - Pays : COUNTRY (F_PAYS), pour les pays la requête récupère le code ISO 2 des pays pour éviter les soucis s'il y a eu un mauvais paramétrage du code ISO (Ex : code Pays FRA pour France mais le code Iso est FR).
 - Conditions de livraison : DELIVERY_COND (P_CONDLIVR)
 - Modes d'expéditions : DELIVERY_MODE (P_EXPEDITION)
 - Fonctions Contacts : CONTACT_POSITION (P_SERVICECPA)
 - Modes de règlement : PAYMENT_MEAN (P_REGLEMENT)

- Codes Taxes Client : TAX_CUSTOMER (F_COMPDET / P_CATCOMPTA)
- Devises.
- Services.
- Catégories comptables.
- Tiers Clients, Prospects.
- Adresses.
- Contacts.
- Articles.
- Tarifs.
- Stocks & Dépôts.
- Historiques et Statistiques de ventes.
- Identifier les différents types d'articles utilisés dans Sage.
 - Articles simples.
 - Articles à gammes.
 - Articles stockés ou Non stockés.
 - CMUP
 - Articles Sérialisés (Lot et Série)
 - Important : les autres types d'articles ne sont pas gérés actuellement (Articles FIFO, LIFO, Articles Ressources)
 - Articles à nomenclatures.
- Identifier les flux de retour de données de Weavy vers Sage 100 cloud.
- Interventions.
- Documents de vente.

C. Déploiement du connecteur Weavy To Sage

1. WeavyToSage

Lancer le setup [WeavyToSage.msi](#).



Cliquer sur Next.

Choisir le serveur sur lequel sera installée la base de données WeavyToSage.

WeavyToSage - InstallShield Wizard

Database Server
Select database server and authentication method

Database server that you are installing to:
[] Browse...

Connect using:
 Windows authentication credentials of current user
 Server authentication using the Login ID and password below

Login ID: [sa]
Password: []

Name of database catalog:
[master] Browse...

InstallShield

< Back Next > Cancel

Cliquer sur Next.

WeavyToSage - InstallShield Wizard

Ready to Install the Program
The wizard is ready to begin installation.

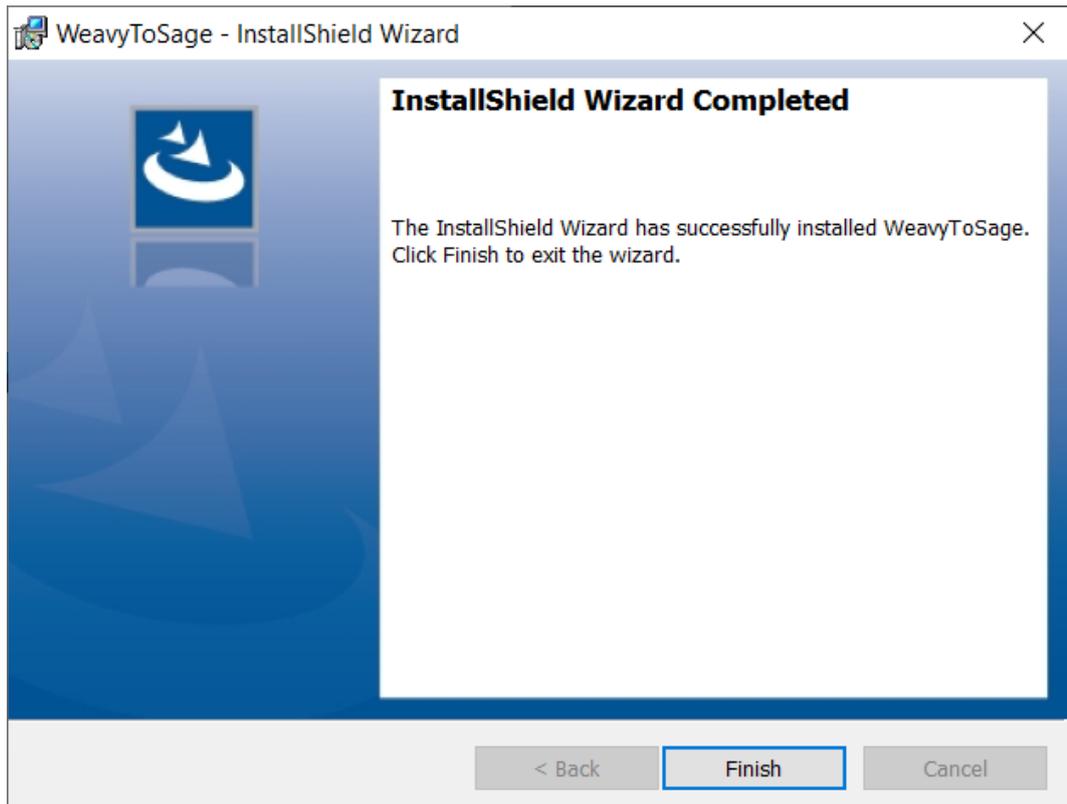
Click Install to begin the installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

InstallShield

< Back Install Cancel

Cliquer sur Install.

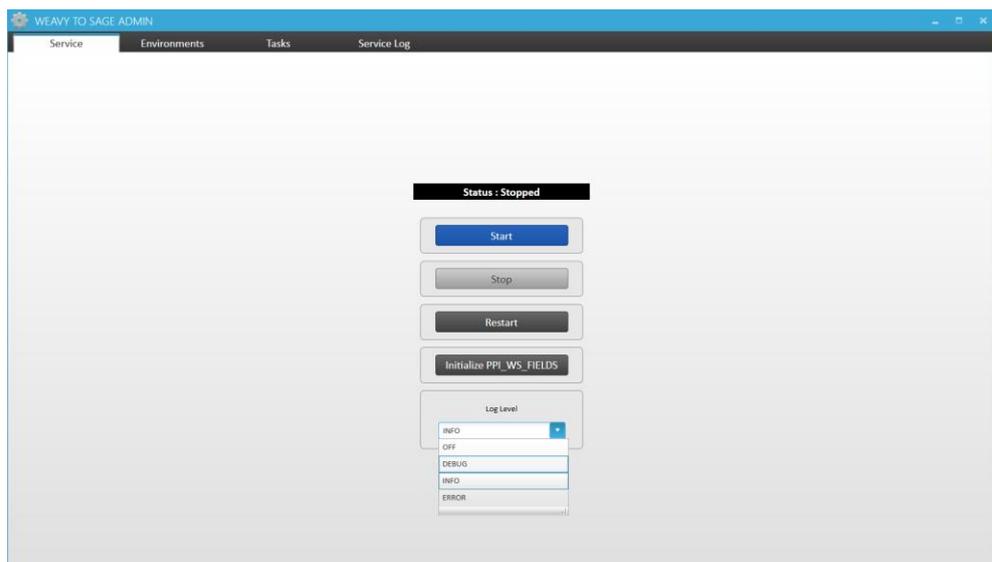


Cliquer sur Finish.

Ensuite il faudra copier la dll « WeavyToSage.Service.dll » du package_install dans le dossier d'installation de WeavyToSage, par défaut « C:\Program Files (x86)\PPI-Group\WeavyToSage ».

I. Service

Ici on peut visualiser l'état du service WeavyToSage, et redémarrer le service Windows



Il est possible de choisir le niveau des logs stockés en base : si un problème d'intégration survient et qu'il nécessite une analyse poussée il est possible de

passer en mode « DEBUG » pour avoir un maximum d'informations. Sinon le niveau « INFO » renseignera largement assez pour chaque tâche.

Le bouton « Initialize PPI_WS_FIELDS » permet d'initialiser tous les champs des tâches existantes aux valeurs par défaut fournies, il est aussi possible d'initialiser les champs par tâches directement, casera vu par la suite.

II. Environnements

Ici il est possible de configurer les différents environnements utilisables pour les synchronisations à venir.



Plusieurs actions sont disponibles sur chaque ligne pour la liste :



: Permet de supprimer l'environnement correspondant.



: Ouvre l'écran de modification ci-dessous.



: Permet la duplication de l'environnement.

Dans l'environnement, il faut spécifier plusieurs choses :

Un nom pour l'environnement.

L'url d'appel de l'API Webservices100, dans laquelle on retrouve le nom de l'environnement API webservices 100.

L'url de l'API Weavy ainsi que le code du projet et les « credentials » (Account name + Password).

On peut aussi remplir les informations pour faire un envoi des rapports par mail.

Note : Pour l'URL de la partie « Sage service », se référer à la documentation d'utilisation de l'API Webservices100 sur <https://webservices100.com/>.

III. Task

Cet onglet permet de paramétrer les différents types de tâches disponibles pour la synchronisation dans WeavyToSage.

Note : L'écran « General information » d'une tâche paramétrée permet de retrouver l'identifiant d'un environnement (Environment Id), qui sera à utiliser dans le paramétrage de la partie [ControlCenter](#).

Active	Send Mail	Detail level	Task Type	Document Filter Type	Label	Environment	Frequency	Description	Last Running	Last Completed	Comment
<input type="checkbox"/>	<input type="checkbox"/>	Detailed error Only	SyncAddress		Your_Environment - Address	Environment Name	Custom	days: 0; hours: 1; minutes: 0			
<input type="checkbox"/>	<input type="checkbox"/>	Detailed error Only	SyncContact		Your_Environment - Contact	Environment Name	Custom	days: 0; hours: 1; minutes: 0			
<input type="checkbox"/>	<input type="checkbox"/>	Detailed error Only	SyncCustomer		Your_Environment - Customer	Environment Name	Custom	days: 0; hours: 1; minutes: 0			
<input type="checkbox"/>	<input type="checkbox"/>	Detailed error Only	SyncInterventions	12000000003 / CLOSED	Your_Environment - Interventions Type 1	Environment Name	Custom	days: 0; hours: 1; minutes: 0			
<input type="checkbox"/>	<input type="checkbox"/>	Detailed error Only	SyncInterventions	12000000004 / CLOSED	Your_Environment - Interventions Type 2	Environment Name	Custom	days: 0; hours: 1; minutes: 0			
<input type="checkbox"/>	<input type="checkbox"/>	Detailed error Only	SyncSalesDocuments	DELIVERY	Your_Environment - SalesDocuments DELIVERY	Environment Name	Custom	days: 0; hours: 1; minutes: 0			
<input type="checkbox"/>	<input type="checkbox"/>	Detailed error Only	SyncSalesDocuments	ORDER	Your_Environment - SalesDocuments ORDERS	Environment Name	Custom	days: 0; hours: 0; minutes: 10			
<input type="checkbox"/>	<input type="checkbox"/>	Detailed error Only	SyncSalesDocuments	QUOTATION	Your_Environment - SalesDocuments QUOTATION	Environment Name	Custom	days: 0; hours: 1; minutes: 0			

EDIT TASK

General information | Mapping | Global Task type fields manager | Task Log

Configuration

Active

Task Type : Sync Address

Label : Your_Environment - Address

Environment : Environment Name (Id: 1)

Frequency : Custom

Days : 0 | Hours : 1 | Minutes : 0

Weavy API Limit : 1000

Weavy API Offset : 0

Weavy API SrvAttrib : 0

Weavy API SrvExport : 1

Send notifications by email

Notification Type : DetailedIfAnyError

Comment

Save | Restore | Close

EDIT TASK

General Information Mapping Global Task type fields manager Task Log

Initialize

Top fields

InsertField	UpdateField	Sage Field	Type	Weavy Field	Default value Sage
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Adresse	string	address1	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CodePostal	string	postalCode	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Commentaire	string	address3	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Complement	string	address2	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Contact	string	contact	
<input type="checkbox"/>	<input type="checkbox"/>	DelaiTransport	int		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Email	string	email	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EstPrincipale	bool	defaultDeliveryAddress	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fax	string	faxNumber	
<input type="checkbox"/>	<input type="checkbox"/>	IdConditionLivraison	short		
<input type="checkbox"/>	<input type="checkbox"/>	IdModeExpedition	short		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Intitule	string	name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Pays	string	genericity_ID_country	

Save Restore Close

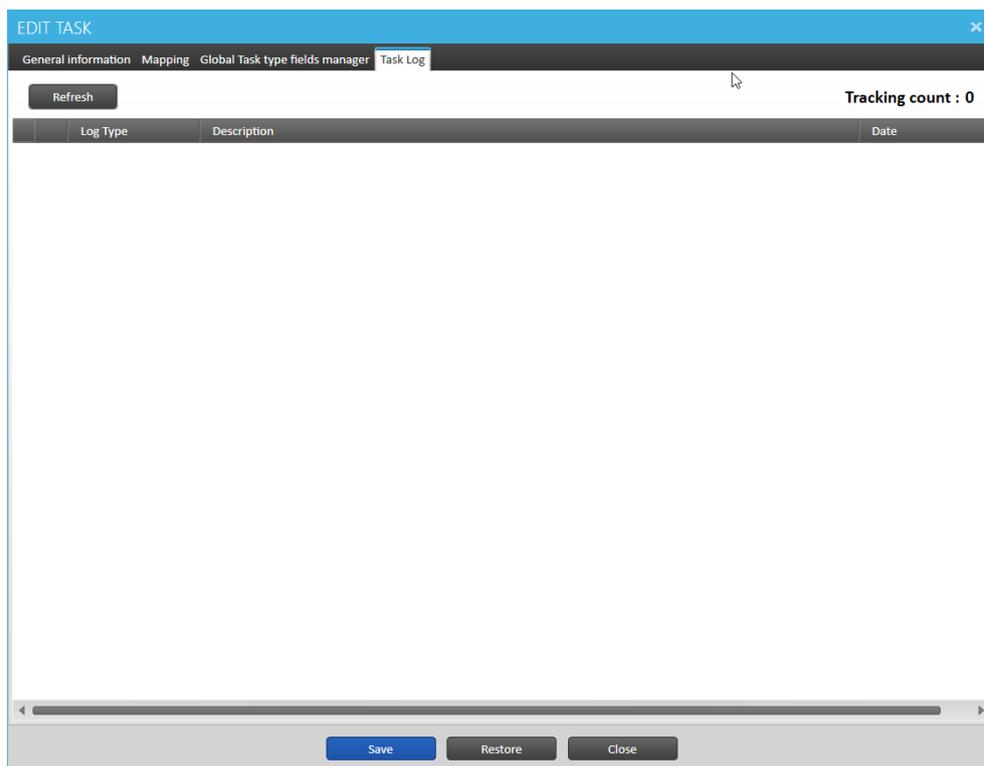
EDIT TASK

General Information Mapping Global Task type fields manager Task Log

PPI_WS Fields content by current task type

FieldId	Task Type	IsWeavy	Field Name	Field Type	Field Level
<input type="checkbox"/>		<input type="checkbox"/>			
137	SyncAddress	<input checked="" type="checkbox"/>	accessCode	Varchar	
119	SyncAddress	<input checked="" type="checkbox"/>	address1	Varchar	
120	SyncAddress	<input checked="" type="checkbox"/>	address2	Varchar	
129	SyncAddress	<input checked="" type="checkbox"/>	address3	Varchar	
145	SyncAddress	<input type="checkbox"/>	Adresse	string	
139	SyncAddress	<input checked="" type="checkbox"/>	apartment	Varchar	
127	SyncAddress	<input checked="" type="checkbox"/>	areaCode	Varchar	
141	SyncAddress	<input checked="" type="checkbox"/>	baseuser_ID_favoriteTechnician	ForeignKey	
132	SyncAddress	<input checked="" type="checkbox"/>	billingAddress	bool	
143	SyncAddress	<input checked="" type="checkbox"/>	branchoffice_ID	ForeignKey	
122	SyncAddress	<input checked="" type="checkbox"/>	city	Varchar	
116	SyncAddress	<input checked="" type="checkbox"/>	CodeCustomeraddress	Varchar	
147	SyncAddress	<input type="checkbox"/>	CodePostal	string	
155	SyncAddress	<input type="checkbox"/>	Commentaire	string	
146	SyncAddress	<input type="checkbox"/>	Complement	string	
128	SyncAddress	<input checked="" type="checkbox"/>	contact	Varchar	
156	SyncAddress	<input type="checkbox"/>	Contact	string	
117	SyncAddress	<input checked="" type="checkbox"/>	customer_ID	ForeignKey	
133	SyncAddress	<input checked="" type="checkbox"/>	defaultBillingAddress	bool	
135	SyncAddress	<input checked="" type="checkbox"/>	defaultDeliveryAddress	bool	
157	SyncAddress	<input type="checkbox"/>	DelaiTransport	int	
134	SyncAddress	<input checked="" type="checkbox"/>	deliveryAddress	bool	

Save Restore Close



Plusieurs actions sont disponibles pour les tâches :

Activer / désactiver une tâche.

Envoyer un mail récapitulatif du traitement avec plusieurs options :

« Summary » : Envoie un mail récapitulatif à la fin de la tâche.

« DetailedIfAnyError » : Envoie un mail récapitulatif + les erreurs concernant cette tâche s'il y en a eu.

« Full » : Envoie un mail contenant un récapitulatif + tous les traitements qui sont affichés dans les logs pour cette tâche.

Un libellé unique.

Choisir l'environnement sur lequel la tâche doit travailler (il ne peut être choisit qu'à la création de la tâche).

Choisir la récurrence d'exécution de chaque tâche.

Choisir les paramètres d'appel concernant l'API Weavy :

« Weavy API Limit » : Nombre de lignes récupérables depuis l'API Weavy (1000 max)

« Weavy API Offset » : Défini à 0, offset du premier élément à renvoyer.

« Weavy API SrvAttrib » : Pour récupérer des lignes avec un SrvAttrib autre que 0.

« Weavy API SrvExport » : Défini à 1. Permet de définir quelles lignes seront récupéré. Une ligne qui a été traitée sera passée à 0 s'il n'y a pas eu d'erreur.

Chaque tâche contient plusieurs onglets :

« General information » : Permet de saisir les informations indiquées ci-dessus ainsi quelques-unes spécifiques en fonction du type de tâche.

« Mapping » : On paramètre dans cet onglet le mapping de correspondance entre les champs Weavy et Sage pour l'insertion et/ou la modification, avec une valeur par défaut si nécessaire.

« Global Task type fields manager » : Permet de rajouter ou de supprimer des champs Sage ou Weavy pour le mapping afin de gérer les évolutions du modèle de données Sage (futures versions) ou Weavy (champs supplémentaires)

A noter que les champs libres Sage, sont gérés de façon native par le connecteur et qu'il n'est pas nécessaire de les ajouter dans cet onglet : il faudra les indiquer directement dans la partie inférieure de l'onglet « Mapping » pour la tâche

« Task Log » : Permet de voir l'historique d'exécution d'une tâche selon le niveau de détail des logs choisi

a. Task SyncAddress

The screenshot shows the 'EDIT TASK' interface for a task named 'Sync Address'. The 'Configuration' tab is selected, displaying various settings for the task. The 'Active' checkbox is unchecked. The 'Task Type' is set to 'Sync Address'. The 'Label' is 'Your_Environment - Address'. The 'Environment' is 'Environment Name (id: 1)'. The 'Frequency' is 'Custom'. The 'Days' field is set to 0, 'Hours' to 1, and 'Minutes' to 0. The 'Weavy API Limit' is 1000, 'Weavy API Offset' is 0, 'Weavy API SrvAttrib' is 0, and 'Weavy API SrvExport' is 1. There is an unchecked checkbox for 'Send notifications by email' and a dropdown for 'Notification Type' set to 'DetailedIfAnyError'. A large 'Comment' text area is on the right. At the bottom, there are 'Save', 'Restore', and 'Close' buttons.

General information Mapping Global Task type fields manager Task Log

Initialize

Top fields

InsertField	UpdateField	Sage Field	Type	Weavy Field	Default value Sage
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Adresse	string	address1	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CodePostal	string	postalCode	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Commentaire	string	address3	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Complement	string	address2	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Contact	string	contact	
<input type="checkbox"/>	<input type="checkbox"/>	DelaiTransport	int		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Email	string	email	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EstPrincipale	bool	defaultDeliveryAddress	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fax	string	faxNumber	
<input type="checkbox"/>	<input type="checkbox"/>	IdConditionLivraison	short		
<input type="checkbox"/>	<input type="checkbox"/>	IdModeExpedition	short		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Intitule	string	name	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Pays	string	generictype_ID_country	
<input type="checkbox"/>	<input type="checkbox"/>	Region	string		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Telephone	string	phoneNumber	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Ville	string	city	

b. Task Contact

EDIT TASK

General information Mapping Global Task type fields manager Task Log

Configuration

Active

Task Type : Sync Contact

Label : Your_Environment - Contact

Environment : Environment Name (id: 1)

Frequency : Custom

Days : Hours : Minutes :

Weavy API Limit :

Weavy API Offset :

Weavy API SrvAttrib :

Weavy API SrvExport :

Send notifications by email

Notification Type : DetailedIfAnyError

Comment

Save Restore Close

General information Mapping Global Task type fields manager Task Log					
Initialize					
Top fields					
InsertField	UpdateField	Sage Field	Type	Weavy Field	Default value Sage
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Civilite	short	generictype_ID_civility	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Email	string	email	
<input type="checkbox"/>	<input type="checkbox"/>	Facebook	string		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fax	string	faxNumber	
<input type="checkbox"/>	<input type="checkbox"/>	Fonction	string		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Gsm	string	mobilePhoneNumber	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IdServiceContact	short	generictype_ID_position	
<input type="checkbox"/>	<input type="checkbox"/>	IdTypeContact	short		
<input type="checkbox"/>	<input type="checkbox"/>	LinkedIn	string		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Nom	string	lastName	
<input type="checkbox"/>	<input type="checkbox"/>	NumeroTiers	string		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Prenom	string	firstName	
<input type="checkbox"/>	<input type="checkbox"/>	Skype	string		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Telephone	string	phoneNumber	

c. Task SyncCustomer

Pour les tâches de types SyncCustomer il est possible de définir les paramètres supplémentaires suivants :

« Is Automatic » : A cocher si la Gestion Commerciale Sage est paramétrée pour avoir une numérotation automatique intégrée (délégation de la numérotation à Sage 100)

« Root Base » + « Third Length » : Si “Is Automatic” n’est pas cochée, permet de préciser la règle de création d’un Code client Sage 100, en définissant une racine alphabétique + une longueur numérique en nombre de digits. Exemple ci-dessous : ‘WV000001’

The screenshot shows the 'EDIT TASK' configuration window for a 'Sync Customer' task. The window is divided into several sections:

- Configuration:**
 - Active
 - Task Type: Sync Customer
 - Label: Your_Environment - Customer
 - Environment: Environment Name (Id: 1)
 - Frequency: Custom
 - Days: 0, Hours: 1, Minutes: 0
 - Weavy API Limit: 1000
 - Weavy API Offset: 0
 - Weavy API SrvAttrib: 0
 - Weavy API SrvExport: 1
 - Send notifications by email
 - Notification Type: DetailedIfAnyError
- Third:**
 - Is Automatic
 - Root Base: WV
 - Third Length: 6
- Comment:** A large empty text area for adding comments.

At the bottom of the window are buttons for 'Save', 'Restore', and 'Close'.

EDIT TASK

General information Mapping Global Task type fields manager Task Log

Initialize

Top fields

InsertField	UpdateField	Sage Field	Type	Weavy Field	Default value Sage
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Adresse	string	address1	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CodePostal	string	postalCode	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Complement	string	address2	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	CompteCollectif	string	accountNumber	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Contact	string	contactName	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Email	string	email	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EstEnSommeil	bool	blocked	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EstProspect	bool	genericType_ID_customerType	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fax	string	faxNumber	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	IdCategorieComptable	int	genericType_ID_accountingCategory	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	IdCategorieTarifaire	short	pricecategory_ID	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	IdDepot	int	productdepot_ID	

Sage free fields

Field	Type	Weavy Field	Default value
-------	------	-------------	---------------

Save Restore Close

Rappel : Dans le cas où il y aurait des champs libres dans Sage, il faudra les ajouter dans la partie inférieure « Sage Free fields » et non pas dans l'onglet « Global Task type fields manager ».

d. Task SyncInterventions

Pour les tâches de types SyncInterventions il est possible de définir les paramètres supplémentaires suivants :

« Intervention Type Filter » : Un filtre sur le type d'intervention Weavy à récupérer (ex : on veut uniquement les tâches de ce type)

« Intervention Status Filter Value » : Un filtre sur le statut de l'intervention à récupérer (ex : on veut uniquement les tâches terminées)

« Sync PDF Report » : Option synchroniser le PDF généré de la fiche d'intervention dans les documents liés de la pièce de ventes Sage.

« Sync Documents » : Option synchroniser les documents de l'intervention dans les documents liés de la pièce de ventes Sage.

« Sync Pictures » : Option synchroniser les images de l'intervention dans les documents liés de la pièce de ventes Sage.

« Range Separator » : caractère séparateur d'échappement pour la gestion des références articles à gammes (voir chapitre dédié dans le Control Center)

The screenshot shows the 'EDIT TASK' configuration window for 'Sync Interventions'. The window is divided into several sections:

- Configuration:**
 - Active
 - Task Type: Sync Interventions
 - Label: Your_Environment - Interventions Type 1
 - Environment: Environment Name (Id: 1)
 - Frequency: Custom
 - Days: 0, Hours: 1, Minutes: 0
 - Weavy API Limit: 100
 - Weavy API Offset: 0
 - Weavy API SrvAttrib: 0
 - Weavy API SrvExport: 1
 - Send notifications by email
 - Notification Type: DetailedIfAnyError
- Filters:**
 - Intervention Type Filter: [Empty dropdown]
 - Intervention Status Filter Value: CLOSED
 - Sync PDF Report
 - Sync Documents
 - Sync Pictures
 - Range separator: -
- Comment:** [Empty text area]

At the bottom of the window are buttons for 'Save', 'Restore', and 'Close'.

Le mapping se sépare en 3 sous-onglets :

« Header » : correspond aux entêtes de documents Gescom Sage

« Detail » : correspond aux lignes de documents Gescom Sage

« Serial » : correspond au détail des n° lot / série des lignes de documents Gescom Sage

EDIT TASK

General Information | **Mapping** | Global Task type fields manager | Task Log

Header | Detail | Serial

Initialize

Sales Document Header fields

Insert	Sage Field	Type	Weavy Field	Default value Sage
<input checked="" type="checkbox"/>	Date	timestamp	startDate	
<input checked="" type="checkbox"/>	IdAdresseLivraison	int	customeraddress_ID	
<input checked="" type="checkbox"/>	IdSouche	short	--EMPTY--	0
<input checked="" type="checkbox"/>	Reference	string	interventionNumber	
<input checked="" type="checkbox"/>	TypeDocument	int	--EMPTY--	2
<input type="checkbox"/>	Contact	string		
<input type="checkbox"/>	CoursDevis	decimal		
<input type="checkbox"/>	DateExpedition	timestamp		
<input type="checkbox"/>	DateLivraison	timestamp		
<input type="checkbox"/>	DateLivraisonRealisee	timestamp		
<input type="checkbox"/>	Donnee1	string		
<input type="checkbox"/>	Donnee2	string		

Sales Document Header Sage free fields

Field	Type	Weavy Field	Default value

Save | Restore | Close

EDIT TASK

General Information | **Mapping** | Global Task type fields manager | Task Log

Header | Detail | Serial

Initialize

Sales Document Details fields

Insert	Sage Field	Type	Weavy Field	Default value Sage
<input checked="" type="checkbox"/>	Codetaxe1	string	tax_ID_1	
<input checked="" type="checkbox"/>	Designation	string	productDescription	
<input checked="" type="checkbox"/>	EstValorise	bool	--EMPTY--	true
<input checked="" type="checkbox"/>	PrixUnitaire	decimal	subPrice	
<input checked="" type="checkbox"/>	Quantite	decimal	quantity	
<input checked="" type="checkbox"/>	RefArticle	string	product_ID	
<input checked="" type="checkbox"/>	RemiseValeur1	decimal	discount	
<input type="checkbox"/>	CodeTaxe2	string		
<input type="checkbox"/>	CodeTaxe3	string		
<input type="checkbox"/>	Conditionnement	string		
<input type="checkbox"/>	DateLivraison	timestamp		
<input type="checkbox"/>	EstNomenclature	bool		

Sales Document Details Sage free fields

Field	Type	Weavy Field	Default value

Save | Restore | Close

EDIT TASK

General information Mapping Global Task type fields manager Task Log

Header Detail Serial

Initialize

Sales Document Serial fields

Insert	Sage Field	Type	Weavy Field	Default value Sage
<input type="checkbox"/>	ComplementLotSerie	string		
<input type="checkbox"/>	DateFabrication	timestamp		
<input checked="" type="checkbox"/>	DatePeremption	timestamp	shelfLife	
<input checked="" type="checkbox"/>	NumLotSerie	string	batchSerialNumber	
<input checked="" type="checkbox"/>	Quantite	decimal	quantity	

Sales Document Serial Sage free fields

Field	Type	Weavy Field	Default value
-------	------	-------------	---------------

Save Restore Close

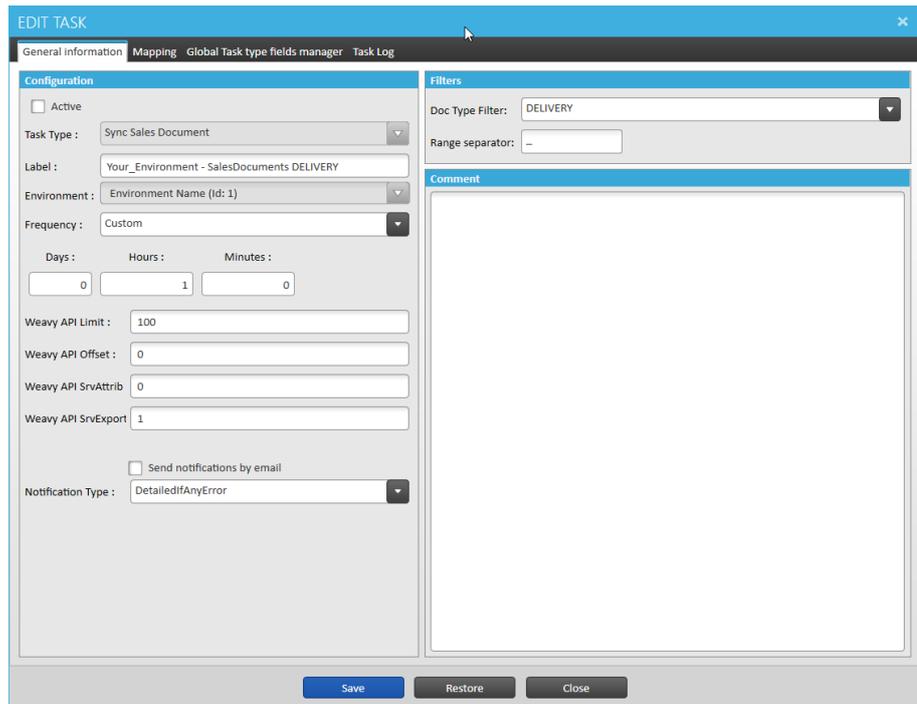
Dans le cas où il y aurait des champs libres dans Sage, il faudra les ajouter dans la partie « Sage Free fields » plutôt que dans « Global Task type fields manager ».

e. [Task SyncSalesDocuments](#)

Pour les tâches de types SyncSalesDocuments il est possible de définir les paramètres supplémentaires suivants :

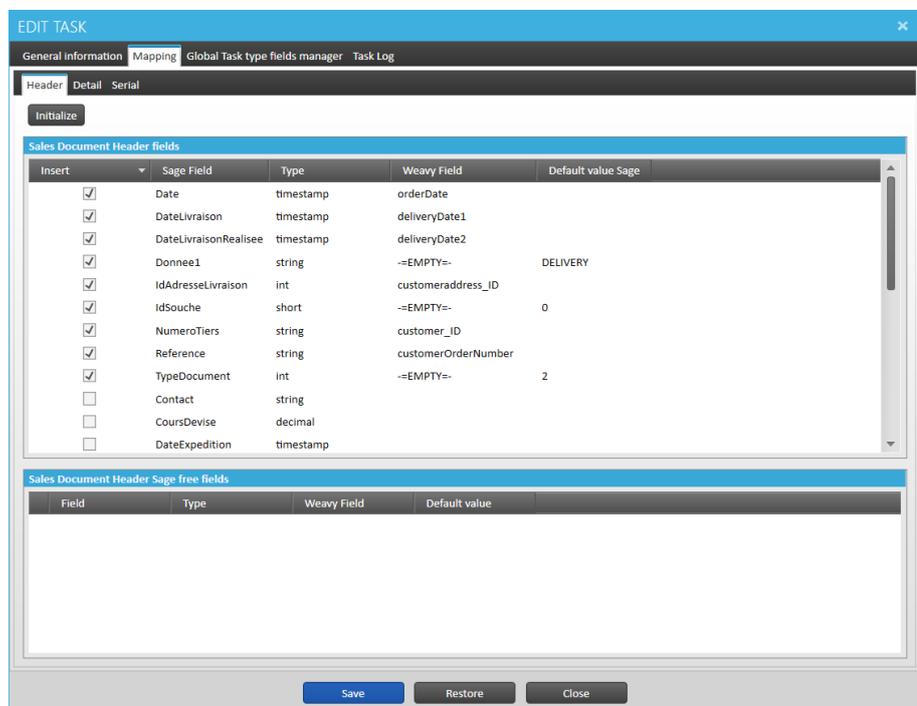
« Doc Type Filter » : Le type de document Weavy à filtrer : Devis, Commande, Livraison

« Range Separator » : caractère séparateur d'échappement pour la gestion des références articles à gammes (voir chapitre dédié dans le Control Center)



Le mapping se sépare en 3 onglets :

- « Header » : correspond aux entêtes de documents Gescom Sage
- « Detail » : correspond aux lignes de documents Gescom Sage
- « Serial » : correspond au détail des n° lot / série des lignes de documents Gescom Sage



EDIT TASK

General Information Mapping Global Task type fields manager Task Log

Header Detail Serial

Initialize

Sales Document Details fields

Insert	Sage Field	Type	Weavy Field	Default value Sage
<input checked="" type="checkbox"/>	CodeTaxe1	string	tax_ID_1	
<input checked="" type="checkbox"/>	Designation	string	productDescription	
<input checked="" type="checkbox"/>	EstValorise	bool	--EMPTY--	true
<input checked="" type="checkbox"/>	PrixUnitaire	decimal	subPrice	
<input checked="" type="checkbox"/>	Quantite	decimal	quantity	
<input checked="" type="checkbox"/>	RefArticle	string	productCode	
<input checked="" type="checkbox"/>	RemiseValeur1	decimal	discount	
<input checked="" type="checkbox"/>	RemiseValeur2	decimal	discount2	
<input checked="" type="checkbox"/>	Taux1	decimal	taxRate1	
<input type="checkbox"/>	CodeTaxe2	string		
<input type="checkbox"/>	CodeTaxe3	string		
<input type="checkbox"/>	Conditionnement	string		

Sales Document Details Sage free fields

Field	Type	Weavy Field	Default value

Save Restore Close

EDIT TASK

General Information Mapping Global Task type fields manager Task Log

Header Detail Serial

Initialize

Sales Document Serial fields

Insert	Sage Field	Type	Weavy Field	Default value Sage
<input type="checkbox"/>	ComplementLotSerie	string		
<input type="checkbox"/>	DateFabrication	timestamp		
<input checked="" type="checkbox"/>	DatePeremption	timestamp	shelfLife	
<input checked="" type="checkbox"/>	NumLotSerie	string	batchSerialNumber	
<input type="checkbox"/>	Quantite	decimal		

Sales Document Serial Sage free fields

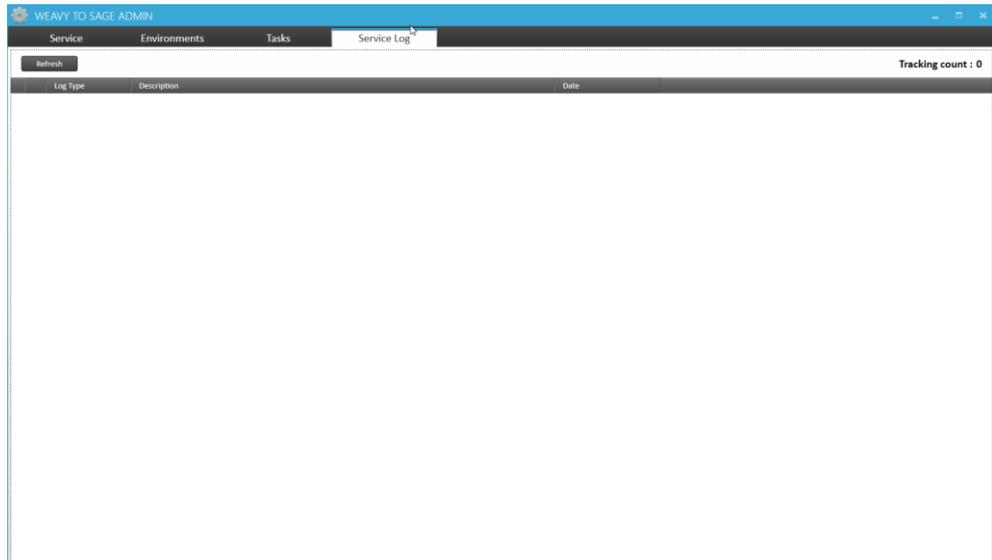
Field	Type	Weavy Field	Default value

Save Restore Close

Dans le cas où il y aurait des champs libres dans Sage, il faudra les ajouter dans la partie « Sage Free fields » plutôt que dans « Global Task type fields manager ».

IV. Service Log

Permet d'accéder aux logs pour toutes les tâches exécutées.



2. Studio

I. Variables

Modifier les variables Weavy suivantes :

The screenshot shows the 'Gestion des variables' window with the following details:

- Search Parameters:** Code variable: SalesCondition, Valeur: (empty), Groupement: (dropdown), Commentaire: (text area).
- Table of Variables:**

Mobile	Code variable	Valeur	Commentaire
<input checked="" type="checkbox"/>	SWB.InterventionParts.SalesCondition.En	Sage	Sales condition engine : Generic, Sage
<input checked="" type="checkbox"/>	SWB.Order.SalesCondition.Engine	Sage	Sales condition engine : Generic, Sage
<input checked="" type="checkbox"/>	SWC.InterventionParts.SalesCondition.En	Sage	Sales condition engine : Generic, Sage
<input checked="" type="checkbox"/>	SWC.Order.SalesCondition.Engine	Sage	Sales condition engine : Generic, Sage
<input checked="" type="checkbox"/>	SWS.InterventionParts.SalesCondition.En	Sage	Sales condition engine : Generic, Sage
<input checked="" type="checkbox"/>	SWS.Order.SalesCondition.Engine	Sage	Sales condition engine : Generic, Sage
- Add Variable Section:**
 - Code variable: SWB.InterventionParts.SalesCondition.Engine
 - Valeur: Sage
 - Groupement: SwingBusiness options
 - Commentaire: Sales condition engine : Generic, Sage
 - Dispo. mobile

Faire une recherche « SalesCondition » dans « Code variable ».

Il faut modifier ces variables pour les passer à la valeur « Sage ».

Ces variables permettent de définir que les règles de tarification appliquées par Weavy sont celle du moteur de calcul Sage (fonctionnalité standard Divalto Weavy).

II. Dictionnaire de données

Il va falloir autoriser certaines actions API pour les tables utilisées par WeavyToSage, à paramétrer depuis le Dictionnaire de données dans le Studio :

Table	API Lecture	API MAJ
sw_data_basedocarchive	X	X
sw_data_basedocdata	X	X
sw_data_customer	X	X
sw_data_customercontact	X	X
sw_data_customeraddress	X	X
sw_data_interventiontype	X	
sw_data_intervention	X	X
sw_data_interventiondocument	X	X
sw_data_interventionpart	X	X
sw_data_interventionpartdetail	X	X
sw_data_interventionpicture	X	X
sw_data_orderheader	X	X
sw_data_orderdetail	X	X
sw_data_orderdetaildetail	X	X

sw_data_productstockqualification	X	X
-----------------------------------	---	---

III. SrvExport

Via le requêteur ou backend il faut mettre à jour toutes les lignes Weavy (Customer, CustomerContact, CustomerAddress) sur ce champ et passer sa valeur à 1 pour le mapping de WeavyToSage.

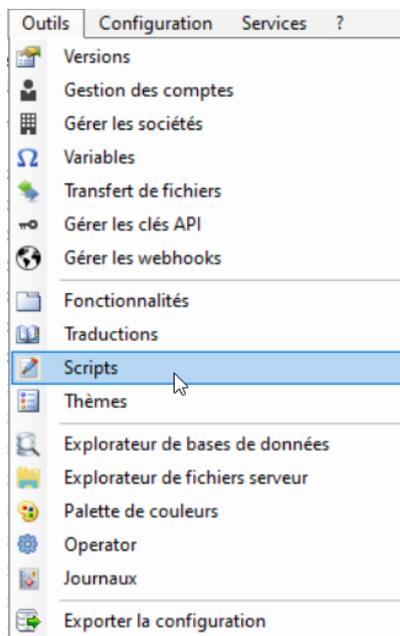
WeavyToSage a besoin d'avoir les tables de mapping (Tiers, Address et Contact) remplies pour fonctionner, sinon il ne pourra pas intégrer les interventions et/ou pièces de vente provenant de Weavy.

IV. Designer

Ici nous allons devoir ajouter des contrôles sur les contacts et adresses car Sage n'accepte pas les doublons d'intitulé.

Dans le Studio, cliquer dans le menu Outils / Scripts.

Aller dans :



Rechercher dans l'explorateur de script Weavy les scripts portant les noms ci-dessous, afin de les remplacer par cette version alternative :

```

Code: FormCustomerContact_Close
Label: Form Customer contact / Close
MASTER - FINAL

1 IF ( CALL_SCRIPT( "Func_IsFormValidated" ) ) THEN
2
3 // Check required fields
4 IF ( FORM_CHECK_REQUIREDFIELDS() == 0 ) THEN
5
6 MSGBOX( "L'opMsg_RequiredFieldsNotFilled/Msg" )
7
8 RETURN( "cancel" )
9
10 ENDF
11
12 // Surcharge WeavyToSage - Début
13 count = VARGET_DBSQL( "SELECT COUNT(*) FROM sw_data_customerContact WHERE customer_ID = 'crownCustomer_ID/row' AND Firstname = 'ctrl:tbFirstname/ctrl:' AND
14 IF ((count > 0) && (EQUALS( "Form_IsValid", "1" ))) THEN
15 MSGBOX( "Contact déjà existant pour ce tiers avec ce nom / prénom" )
16 RETURN( "cancel" )
17 ENDF
18 // Surcharge WeavyToSage - Fin
19
20 // check email
21 IF ( LEN( "ctrl:tbEmail/ctrl:" ) > 0 ) THEN
22 emailOK = CALL_SCRIPT( "Func_CheckEmail", "ctrl:tbEmail/ctrl:" )
23
24 IF ( EQUALS( emailOK, "1" ) == 0 ) THEN
25 MSGBOX( "L'opMsg_InvalidEmail/Msg" )
26 RETURN( "cancel" )
27 ENDF
28
29 ENDF
30
31
32 // Save data
33 FORM_SAVE_DATA()
34
35
36 // Add consent event and customer contact extension
37 IF ( EQUALS( "Form_IsValid", "1" ) ) THEN
38 // Add consent event
39 eventLabel = VARGET_DBSQL( "SELECT TRIM( IFNULL( firstname, '' ) || ' ' || IFNULL( lastname, '' ) ) FROM sw_data_customerContact WHERE customercontact_ID = "
40 event_ID = CALL_SCRIPT( "Func_AdaptateEvent", "Consent", "", "sw_data_customer_ID/row", "sw_data_customercontact_ID/row", "" )
41 // Add customer contact extension
42 VARGET_DB( "sw_data_customercontactextension", "customercontact_RETURN: the formatted current date and time" )
43 ENDF
44
45
46 // //update contact on device
47 // If contacts synchronization is active
48 IF ( CALL_SCRIPT( "Func_GetDeviceContactsAccess" ) ) THEN
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

La partie des scripts modifiée pour les besoins du connecteur WeavyToSage est identifiable par les balises de commentaires « //Surcharge WeavyToSage – Début » et « //Surcharge WeavyToSage – Fin »

IMPORTANT : Ces scripts s’appuient sur la version standard des scripts de Weavy. Si vous avez déjà réalisé des modifications des versions standards de ces scripts, vous devez adapter le code contenu entre nos balises et tester le bon fonctionnement.

a. [Contact](#)

Ouvrir le script [FormCustomerContact Close](#).

Cette modification permet d’éviter d’avoir un doublon de contact qui ont le même nom et prénom pour un client.

b. [Adresses](#)

Ouvrir le script [FormCustomerAddress Close](#).

Cette modification permet d’éviter d’avoir un doublon d’intitulé pour le même client.

c. [SalesCondition_Calculate_Sage](#)

Ouvrir le script [SalesCondition Calculate Sage](#).

Reprend les calculs de prix par rapport à Sage et l’import du ControlCenter.

d. [FormBatchSerialNumbersSelectionusBatchSerialNumbers_ValueChanged](#)

Ouvrir le script

[FormBatchSerialNumbersSelectionusBatchSerialNumbers ValueChanged](#).

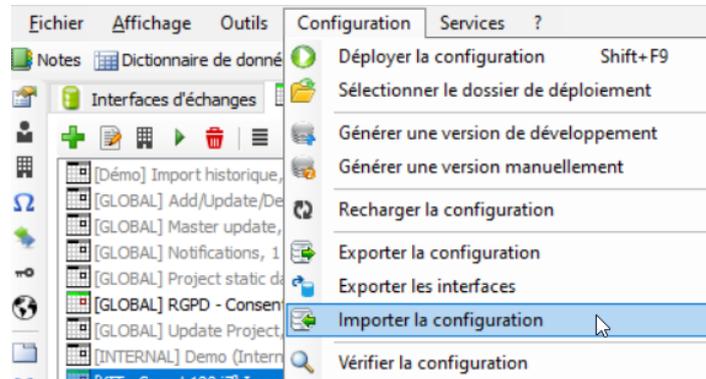
Ici on va vérifier qu’il y a toujours du stock de disponible.

V. Backend

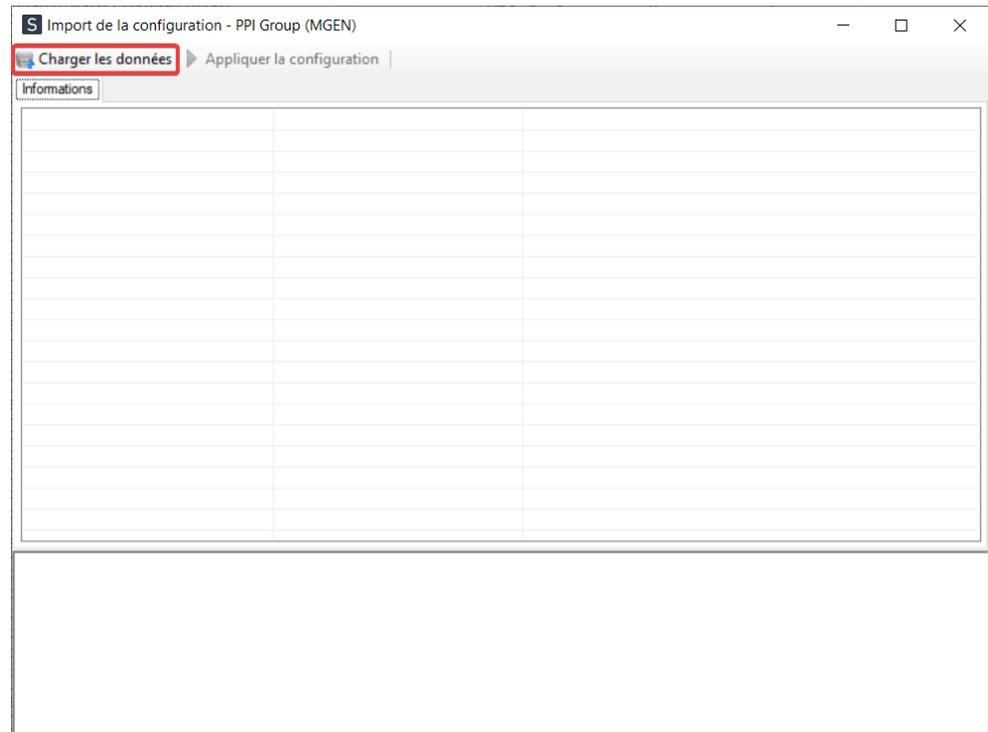
Il faudra ajouter plusieurs backends et une planification pour ces backends qui pourra être reprise (champ Backend automate ID) dans le Control Center pour un lancement automatique.

Note : attention le mapping n'est possible que dès lors que vous avez généré les fichiers CSV depuis le Control Center et que ces fichiers ont été transférés dans le cloud Weavy via le module approprié (Voir chapitre [Control Center](#))

Vous pouvez aussi importer directement les interfaces en utilisant l'option d'import :



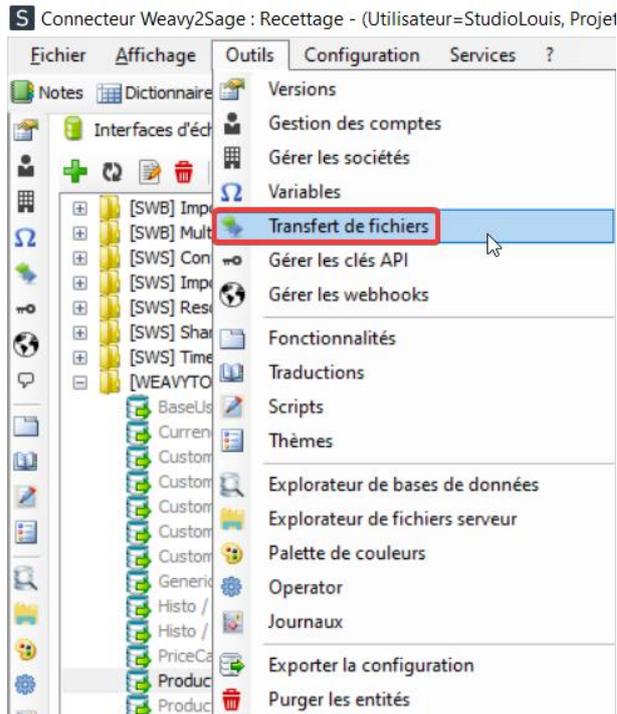
Cliquer sur « Importer la configuration ».



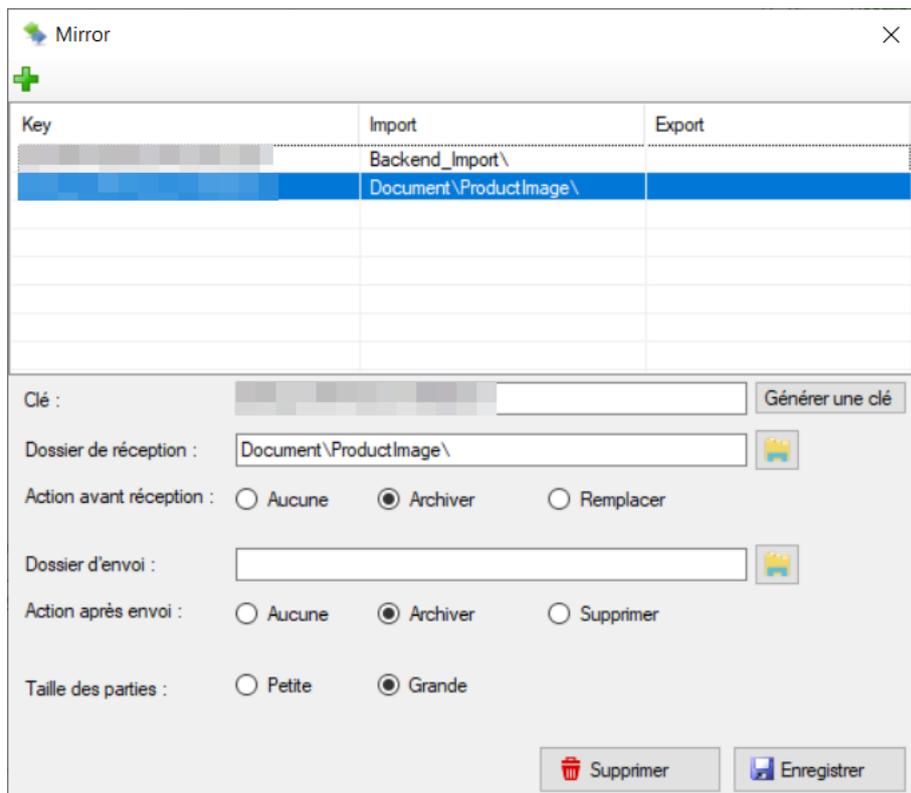
Cliquer ensuite dans « Charger les données ».

Il vous faudra aller chercher le dossier /Studio/interface backend.

On va aussi générer les clés qui seront utilisés par le ControlCenter par la suite :



On va générer une clé pour faire l'import des images et une autre pour faire l'import des .csv comme ceci :



a. Backend Base User (sw_data_baseuser)

Interface :

Mapping :

BaseUser BaseUser.csv		Base / User (sw_data_baseuser)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	codeBASEuser	0	Base / User		Varchar (255)	CodeBaseuser	1
1	lastName	1	Last Name		Varchar (64)	lastName	2
2	firstName	2	First Name		Varchar (64)	firstName	3
3	Initials	3	Initials		Varchar (64)	initials	4
4	email	4	Email		Varchar (255)	email	5
5	phoneNumber	5	Phone number		Varchar (64)	phoneNumber	6
6	faxNumber	6	Fax number		Varchar (64)	faxNumber	7
7	address1	7	Address 1		Varchar (64)	address1	8
8	address2	8	Address 2		Varchar (64)	address2	9
9	postalCode	9	Postal code		Varchar (10)	postalCode	10
10	city	10	City		Varchar (64)	city	11
11	position	11	Position		Varchar (64)	position	13
12	department	12	Department		Varchar (64)	department	16
13	genericType_ID_country	13	Country		ForeignKey	genericType_ID_country	38

b. Backend Currency (sw_data_currency)

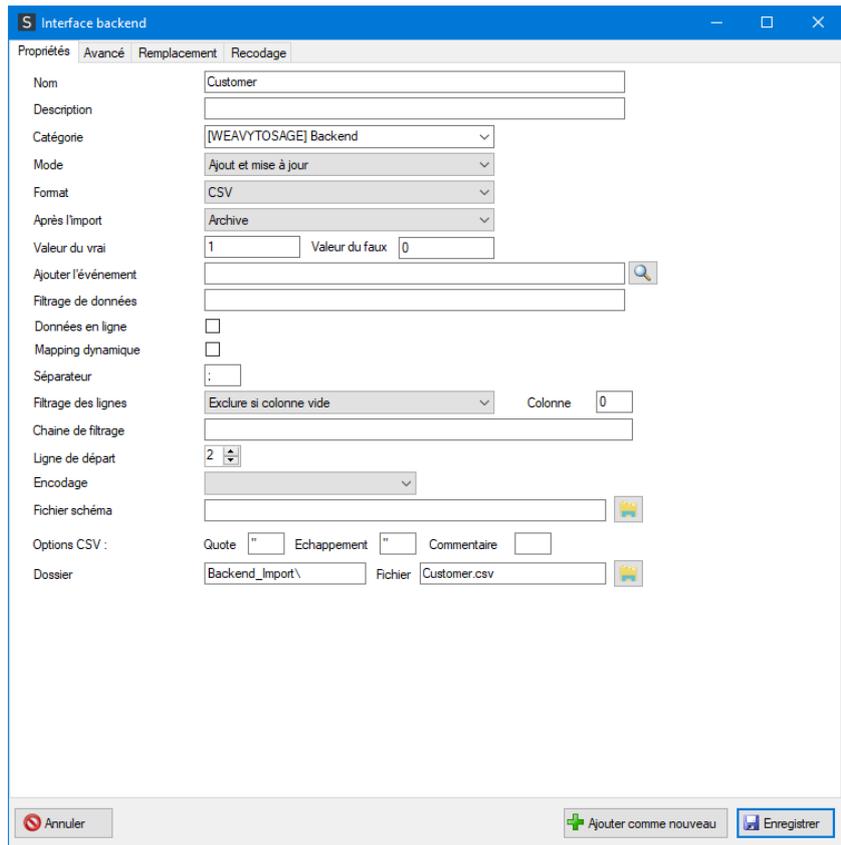
Interface :

Mapping :

Currency		Currency (sw_data_currency)					
Currency.csv		⇒* Auto mapping					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	CodeCurrency	0	Currency		Varchar (255)	CodeCurrency	1
1	label	1	Label		Varchar (64)	label	2
2	format	2	Format		Varchar (64)	format	3
3	currencyRate	3	Currency rate		Double	currencyRate	4
4	periodRate	4	Course period		Varchar (64)	periodRate	5
5	currencyUnit	5	Currency unit		Varchar (64)	currencyUnit	6
6	currencySubunit	6	Currency sub-unit		Varchar (64)	currencySubunit	7
7	iso	7	ISO		Varchar (64)	iso	8

c. Backend Customer (sw_data_customer)

Interface :



Mapping :

Customer		Customer (sw_data_customer)					
Customer.csv		Auto mapping					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	CodeCustomer	0	Account code		Varchar (255)	CodeCustomer	1
1	baseuser_ID	1	Base / User		ForeignKey	baseuser_ID	2
2	name	2	Name		Varchar (255)	name	4
3	address1	3	Address 1		Varchar (255)	address1	5
4	address2	4	Address 2		Varchar (255)	address2	6
5	postalCode	5	Postal code		Varchar (16)	postalCode	7
6	city	6	City		Varchar (64)	city	8
7	phoneNumber	7	Phone number		Varchar (64)	phoneNumber	9
8	faxNumber	8	Fax number		Varchar (64)	faxNumber	10
9	email	9	Email		Varchar (255)	email	11
10	contactName	10	Contact name		Varchar (64)	contactName	12
11	website	11	Website		Varchar (255)	website	13
12	siretNumber	12	TPS / TVQ		Varchar (40)	siretNumber	21
13	message	13	Message		Text (65535)	message	28
14	blocked	14	Blocked		Bool	blocked	30
15	underSurveillance	15	Under surveillance		Bool	underSurveillance	31
16	pricecategory_ID	16	Price Category		ForeignKey	pricecategory_ID	49
17	discountRate	17	Discount rate		Double	discountRate	50
18	apeCode	18	APE code		Varchar (64)	apeCode	78
19	vatIntra	19	Intra-Community VAT		Varchar (64)	vatIntra	79
20	legalStatus	20	Forme juridique		Varchar (64)	legalStatus	80
21	workforce	21	Workforce		Int	workforce	81
22	customer_ID_paying	22	Paying customer		ForeignKey	customer_ID_paying	82
23	risk	23	Risk		Varchar (64)	risk	87
24	language	24	Language		Int	language	88
25	accountNumber	25	Main account number		Varchar (64)	accountNumber	89
26	currency_ID	26	Currency		ForeignKey	currency_ID	97
27	generictype_ID_customerFamily	27	Third Party family		ForeignKey	generictype_ID_customerFamily	32
28	generictype_ID_customerType	28	Account type		ForeignKey	generictype_ID_customerType	33
29	generictype_ID_paymentMean	29	Payment mean		ForeignKey	generictype_ID_paymentMean	35
30	generictype_ID_deliveryMode	30	Delivery mode		ForeignKey	generictype_ID_deliveryMode	36
31	generictype_ID_country	31	Country		ForeignKey	generictype_ID_country	72
32	generictype_ID_taxCode	32	Customer tax code		ForeignKey	generictype_ID_taxCode	84

d. Backend Customer / Address (sw_data_customeraddress)

Interface :

Mapping :

Customer / Address Customer_Address.csv		Customer / Address (sw_data_customeraddress)					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	CodeCustomeraddress	0	Customer / Address		Varchar (255)	CodeCustomeraddress	1
1	customer_ID	1	Customer		ForeignKey	customer_ID	2
2	name	2	Name		Varchar (255)	name	3
3	address1	3	Address 1		Varchar (255)	address1	4
4	address2	4	Address 2		Varchar (255)	address2	5
5	postalCode	5	Postal code		Varchar (16)	postalCode	6
6	city	6	City		Varchar (64)	city	7
7	phoneNumber	7	Phone number		Varchar (64)	phoneNumber	8
8	faxNumber	8	Fax number		Varchar (64)	faxNumber	9
9	email	9	Email		Varchar (255)	email	11
10	contact	10	Contact		Varchar (64)	contact	15
11	areaCode	11	Area code		Varchar (64)	areaCode	13
12	billingAddress	12	Billing address		Bool	billingAddress	24
13	defaultBillingAddress	13	Default billing address		Bool	defaultBillingAddress	25
14	deliveryAddress	14	Delivery address		Bool	deliveryAddress	26
15	DefaultDeliveryAddress	15	Default delivery address		Bool	defaultDeliveryAddress	27
16	site	16	Site		Bool	site	28
17	genericType_ID_country	17	Country		ForeignKey	genericType_ID_country	21

- e. Backend Customer / Contact (sw_data_customercontact)

Interface :

Mapping :

Customer / Contact		Customer / Contact (sw_data_customercontact)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	CodeCustomercontact	0	Customer / Contact		Varchar (255)	CodeCustomercontact	1
1	customer_ID	1	Customer		ForeignKey	customer_ID	2
2	lastName	2	Last Name		Varchar (255)	lastName	5
3	firstName	3	First Name		Varchar (255)	firstName	6
4	phoneNumber	4	Phone number		Varchar (64)	phoneNumber	11
5	mobilePhoneNumber	5	Mobile number		Varchar (64)	mobilePhoneNumber	12
6	faxNumber	6	Fax number		Varchar (64)	faxNumber	13
7	email	7	Email		Varchar (255)	email	14
8	description	8	Description		Varchar (255)	description	16
9	genericType_ID_civility	9	Civility		ForeignKey	genericType_ID_civility	4
10	genericType_ID_position	10	Position		ForeignKey	genericType_ID_position	17

f. Backend Customer Credit (sw_data_customercredit)

Interface :

Mapping :

Customer / Credit Customer_Credit.csv		Customer / Credit (sw_data_customercredit)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	CodeCustomerCredit	0	Customer / Credit		Varchar (255)	CodeCustomercredit	1
1	customer_ID	1	Customer		ForeignKey	customer_ID	2
2	documentNumber	2	Document number		Varchar (64)	documentNumber	3
3	documentDate	3	Document date		Timestamp	documentDate	4
4	limitDate	4	Limit date		Timestamp	limitDate	5
5	amount	5	Amount		Double	amount	6
6	documentType	6	Document type		Varchar (64)	documentType	7
7	label	7	Label		Varchar (255)	label	8
8	period	8	Période		DateTime	period	9

g. Backend Customer / User (sw_data_customeruser)

Interface :

Mapping :

Customer / User Customer_User.csv		Customer / User (sw_data_customeruser)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	customer_ID	0	Customer		ForeignKey	customer_ID	2
1	baseuser_ID	1	Base / User		ForeignKey	baseuser_ID	3
2	assignmentFromSwing	2	Assignment from Divalto ...		Bool	assignmentFromSwing	6
3	CodeCustomerUser	3	Customer / User		Varchar (255)	CodeCustomeruser	1

h. Backend GenericType (sw_data_generictype)

Interface :

Mapping :

Generic Type GenericType.csv		Generic type (sw_data_generictype)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	originalCode	0	Original code		Varchar (255)	originalCode	6
1	label	1	Label		Varchar (255)	label	2
2	dataType	2	Data type		Varchar (64)	dataType	3
3	codeGenerictype	3	Code		Varchar (255)	codeGenerictype	1

- i. Backend Histo Detail (sw_data_histodetail)

Interface :

Mapping :

Histo / Detail		History / Detail (sw_data_histodetail)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	histoheader_ID	0	History / Header		ForeignKey	histoheader_ID	2
1	customer_ID	1	Customer		ForeignKey	customer_ID	3
2	lineNumber	2	Line number		Int	lineNumber	4
3	product_ID	3	Product		ForeignKey	product_ID	5
4	orderedQuantity	4	Ordered quantity		Double	orderedQuantity	6
5	deliveredQuantity	5	Delivered quantity		Double	deliveredQuantity	7
6	docNumber	6	Document number		Varchar (64)	docNumber	8
7	docDate	7	Document date		Timestamp	docDate	9
8	amount	8	Amount		Double	amount	10
9	price	9	Price		Double	price	12
10	externalDocNumber	10	External document number		Varchar (64)	externalDocNumber	15
11	discountValue1	11	Discount value 1		Double	discountValue1	18
12	discountValue2	12	Discount value 2		Double	discountValue2	20
13	discountValue3	13	Discount value 3		Double	discountValue3	22
14	deliveryDate	14	Delivery date		Timestamp	deliveryDate	24
15	customerProductRef	15	Customer product reference		Varchar (255)	customerProductRef	25
16	productNotDelivered	16	Product not delivered		Bool	productNotDelivered	26
17	productDescription	17	Product description		Varchar (255)	productDescription	27
18	productCode	18	Product code		Varchar (255)	productCode	29
19	baseuser_ID	19	Base / User		ForeignKey	baseuser_ID	30
20	CodeHistodetail	20	History / Line		Varchar (255)	CodeHistodetail	1

j. Backend Histo Header (sw_data_histoheader)

Interface :

Mapping :

Histo / Header		History / Header (sw_data_histoheader)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	customer_ID	0	Customer		ForeignKey	customer_ID	2
1	CodeHistoheader	1	History / Header		Varchar (255)	CodeHistoheader	1
2	docNumber	2	Document number		Varchar (64)	docNumber	3
3	docDate	3	Document date		Timestamp	docDate	4
4	externalDocNumber	4	External document number		Varchar (64)	externalDocNumber	5
5	totalAmount	5	Total amount		Double	totalAmount	6
6	remainingDocument	6	Backorder document		Bool	remainingDocument	8
7	deliveryDate	7	Delivery date		Timestamp	deliveryDate	9
8	priceCategory_ID	8	Price Category		ForeignKey	pricecategory_ID	10
9	discountFoot	9	Discount foot document		Double	discountFoot	17
10	baseuser_ID	10	Base / User		ForeignKey	baseuser_ID	20
11	generictype_ID_doctype	11	Document Type		ForeignKey	generictype_ID_doctype	18

k. Backend PriceCategory (sw_data_pricecategory)

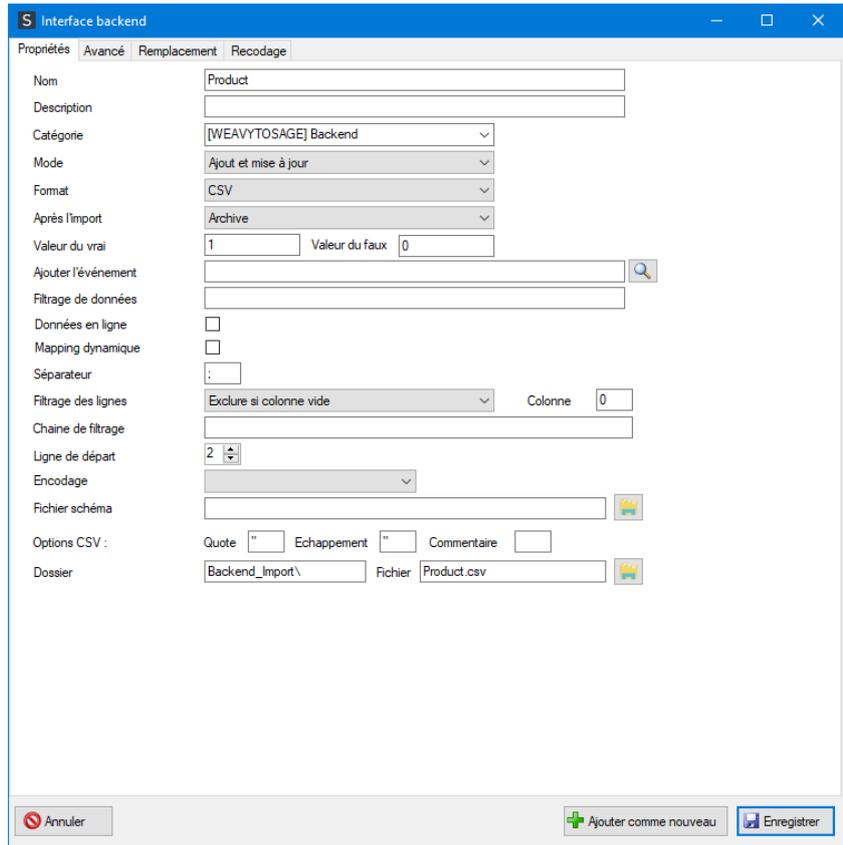
Interface :

Mapping :

PriceCategory PriceCategory.csv		Tarif / Catégorie tarifaire (sw_data_pricecategory)					
Pos.	Champ	Pos.	Champ	Propriétés	Type	Nom	Diction...
0	CodePricecategory	0	Tarif / Catégorie tarifaire		Varchar (255)	CodePricecategory	1
1	label	1	Libellé		Varchar (255)	label	2

I. Backend Product (sw_data_product)

Interface :

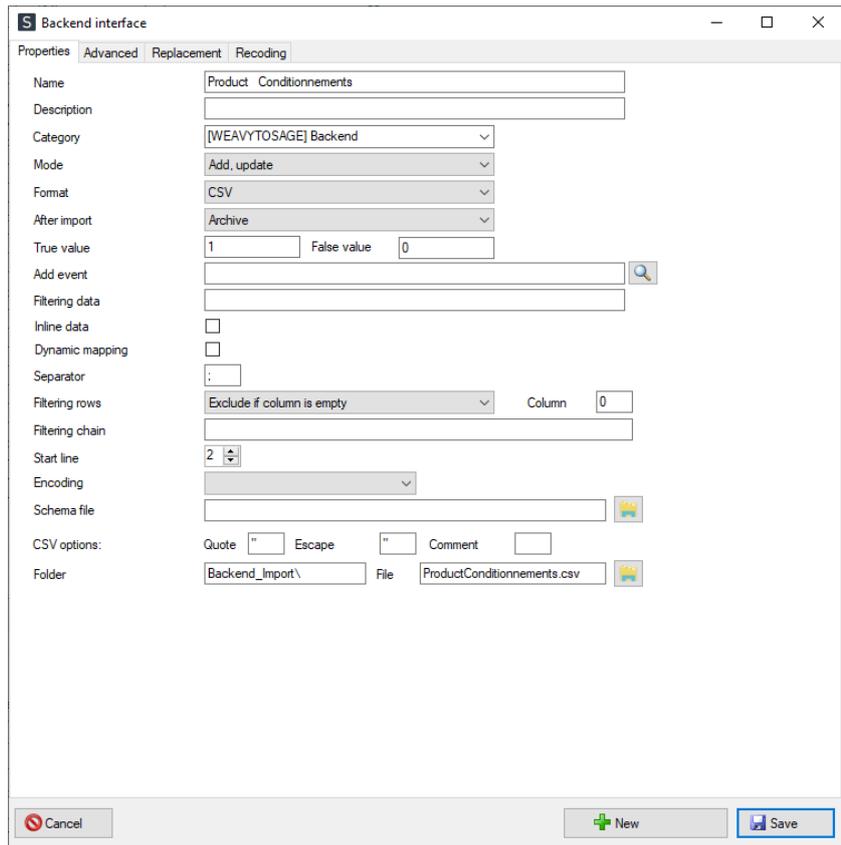


Mapping :

Product		Product (sw_data_product)					
Product.csv				Auto mapping			
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	CodeProduct	0	Product		Varchar (255)	CodeProduct	1
1	description	1	Description		Varchar (255)	description	2
2	productfamily_ID	2	Product / Family		ForeignKey	productfamily_ID	4
3	saleMultiple	3	Sale multiple		Double	saleMultiple	5
4	defaultPrice	4	Default price		Double	defaultPrice	6
5	purchasePrice	5	Purchase price		Double	purchasePrice	7
6	coefficient	6	Coefficient		Double	coefficient	9
7	weight	7	Weight		Double	weight	10
8	barcode	8	Barcode		Varchar (64)	barcode	14
9	packaging	9	Packaging		Varchar (64)	packaging	22
10	supplierReference	10	Supplier reference		Varchar (64)	supplierReference	32
11	supplierBarcode	11	Supplier barcode		Varchar (64)	supplierBarcode	33
12	blocked	12	Blocked		Bool	blocked	47
13	supplier_ID	13	Supplier		ForeignKey	supplier_ID	54
14	supplementaryDescription1	14	Supplementary description 1		Text (65535)	supplementaryDescription1	55
15	supplementaryDescription2	15	Supplementary description 2		Text (65535)	supplementaryDescription2	56
16	unused	16	Unused		Bool	unused	58
17	height	17	Height		Double	height	60
18	width	18	Width		Double	width	61
19	depth	19	Depth		Double	depth	62
20	productGeneric	20	Generic product		Bool	productGeneric	65
21	product_ID_parentDeclination	21	Parent product for declina...		ForeignKey	product_ID_parentDeclination	66
22	productdeclination_ID_level1	22	Variant / Level 1		ForeignKey	productdeclination_ID_level1	67
23	productdeclination_ID_level2	23	Variant / Level 2		ForeignKey	productdeclination_ID_level2	68
24	stockDeclination	24	Stock management with v...		Bool	stockDeclination	72
25	erpProductID	25	ERP Product ID		Varchar (255)	erpProductID	73
26	generictype_ID_weightunit	26	Weight unit		ForeignKey	generictype_ID_weightunit	13
27	generictype_ID_taxCode	27	Product tax code		ForeignKey	generictype_ID_taxCode	23
28	generictype_ID_sellingUnit	28	Selling unit		ForeignKey	generictype_ID_sellingUnit	57
29	generictype_ID_managementType	29	Management type		ForeignKey	generictype_ID_managementType	59
30	generictype_ID_lengthunit	30	Length unit		ForeignKey	generictype_ID_lengthunit	63
31	generictype_ID_brand	31	Brand		ForeignKey	generictype_ID_brand	64
32	final_w2s_FactPoids	32	final_w2s_FactPoids		Bool	final_w2s_FactPoids	75

m. Backend Product Conditionnement (sw_data_product)

Interface :

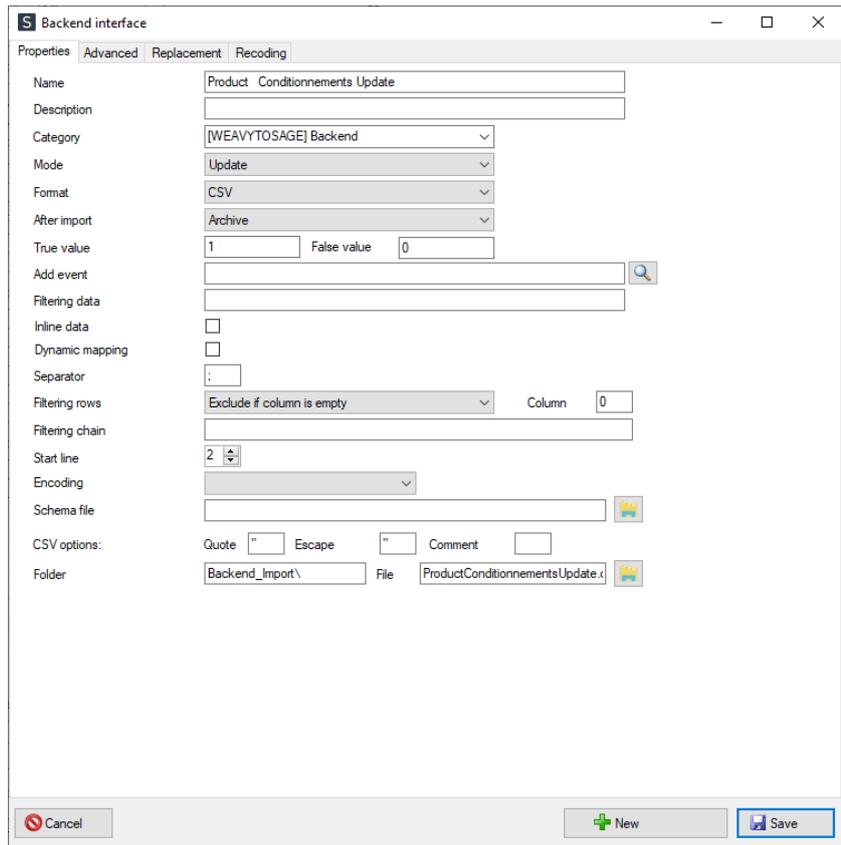


Mapping :

Product Conditionnements ProductConditionnements.csv		Product (sw_data_product) Auto mapping					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	CodeProduct	0	Product		Varchar (255)	CodeProduct	1
1	description	1	Description		Varchar (255)	description	2
2	productfamily_ID	2	Product / Family		ForeignKey	productfamily_ID	4
3	saleMultiple	3	Sale multiple		Double	saleMultiple	5
4	defaultPrice	4	Default price		Double	defaultPrice	6
5	purchasePrice	5	Purchase price		Double	purchasePrice	7
6	coefficient	6	Coefficient		Double	coefficient	9
7	weight	7	Weight		Double	weight	10
8	barcode	8	Barcode		Varchar (64)	barcode	14
9	packaging	9	Packaging		Varchar (64)	packaging	22
10	supplierReference	10	Supplier reference		Varchar (64)	supplierReference	32
11	supplierBarcode	11	Supplier barcode		Varchar (64)	supplierBarcode	33
12	blocked	12	Blocked		Bool	blocked	47
13	supplier_ID	13	Supplier		ForeignKey	supplier_ID	54
14	supplementaryDescription1	14	Supplementary description 1		Text (65535)	supplementaryDescription1	55
15	supplementaryDescription2	15	Supplementary description 2		Text (65535)	supplementaryDescription2	56
16	unused	16	Unused		Bool	unused	58
17	height	17	Height		Double	height	60
18	width	18	Width		Double	width	61
19	depth	19	Depth		Double	depth	62
20	productGeneric	20	Generic product		Bool	productGeneric	65
21	product_ID_parentDeclination	21	Parent product for declina...		ForeignKey	product_ID_parentDeclination	66
22	productdeclination_ID_level1	22	Variant / Level 1		ForeignKey	productdeclination_ID_level1	67
23	productdeclination_ID_level2	23	Variant / Level 2		ForeignKey	productdeclination_ID_level2	68
24	stockDeclination	24	Stock management with v...		Bool	stockDeclination	72
25	erpProductID	25	ERP Product ID		Varchar (255)	erpProductID	73
26	generictype_ID_weightunit	26	Weight unit		ForeignKey	generictype_ID_weightunit	13
27	generictype_ID_taxCode	27	Product tax code		ForeignKey	generictype_ID_taxCode	23
28	generictype_ID_sellingUnit	28	Selling unit		ForeignKey	generictype_ID_sellingUnit	57
29	generictype_ID_managementType	29	Management type		ForeignKey	generictype_ID_managementType	59
30	generictype_ID_lengthunit	30	Length unit		ForeignKey	generictype_ID_lengthunit	63
31	generictype_ID_brand	31	Brand		ForeignKey	generictype_ID_brand	64
32	final_w2s_FactPoids	32	final_w2s_FactPoids		Bool	final_w2s_FactPoids	75

n. Backend Product Conditionnements update (sw_data_product)

Interface :

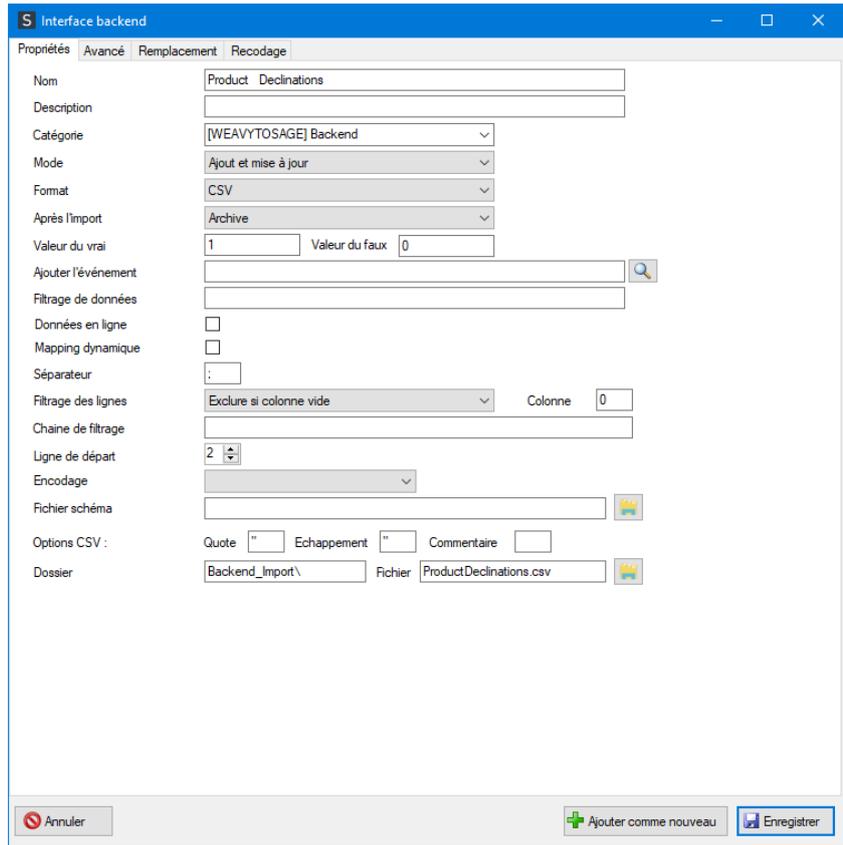


Mapping :

Product Conditionnements Update		Product (sw_data_product)					
ProductConditionnementsUpdate.		Product (sw_data_product)		⇒ Auto mapping			
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	CodeProduct	0	Product		Varchar (255)	CodeProduct	1
1	description	1	Description		Varchar (255)	description	2
2	productfamily_ID	2	Product / Family		ForeignKey	productfamily_ID	4
3	saleMultiple	3	Sale multiple		Double	saleMultiple	5
4	defaultPrice	4	Default price		Double	defaultPrice	6
5	purchasePrice	5	Purchase price		Double	purchasePrice	7
6	coefficient	6	Coefficient		Double	coefficient	9
7	weight	7	Weight		Double	weight	10
8	barcode	8	Barcode		Varchar (64)	barcode	14
9	packaging	9	Packaging		Varchar (64)	packaging	22
10	supplierReference	10	Supplier reference		Varchar (64)	supplierReference	32
11	supplierBarcode	11	Supplier barcode		Varchar (64)	supplierBarcode	33
12	blocked	12	Blocked		Bool	blocked	47
13	supplier_ID	13	Supplier		ForeignKey	supplier_ID	54
14	supplementaryDescription1	14	Supplementary description 1		Text (65535)	supplementaryDescription1	55
15	supplementaryDescription2	15	Supplementary description 2		Text (65535)	supplementaryDescription2	56
16	unused	16	Unused		Bool	unused	58
17	height	17	Height		Double	height	60
18	width	18	Width		Double	width	61
19	depth	19	Depth		Double	depth	62
20	productGeneric	20	Generic product		Bool	productGeneric	65
21	product_ID_parentDeclination	21	Parent product for declina...		ForeignKey	product_ID_parentDeclination	66
22	productdeclination_ID_level1	22	Variant / Level 1		ForeignKey	productdeclination_ID_level1	67
23	productdeclination_ID_level2	23	Variant / Level 2		ForeignKey	productdeclination_ID_level2	68
24	stockDeclination	24	Stock management with v...		Bool	stockDeclination	72
25	erpProductID	25	ERP Product ID		Varchar (255)	erpProductID	73
26	generictype_ID_weightunit	26	Weight unit		ForeignKey	generictype_ID_weightunit	13
27	generictype_ID_taxCode	27	Product tax code		ForeignKey	generictype_ID_taxCode	23
28	generictype_ID_sellingUnit	28	Selling unit		ForeignKey	generictype_ID_sellingUnit	57
29	generictype_ID_managementType	29	Management type		ForeignKey	generictype_ID_managementType	59
30	generictype_ID_lengthunit	30	Length unit		ForeignKey	generictype_ID_lengthunit	63
31	generictype_ID_brand	31	Brand		ForeignKey	generictype_ID_brand	64
32	final_w2s_FactPoids	32	final_w2s_FactPoids		Bool	final_w2s_FactPoids	75

o. Backend Product Declinations (sw_data_product)

Interface :

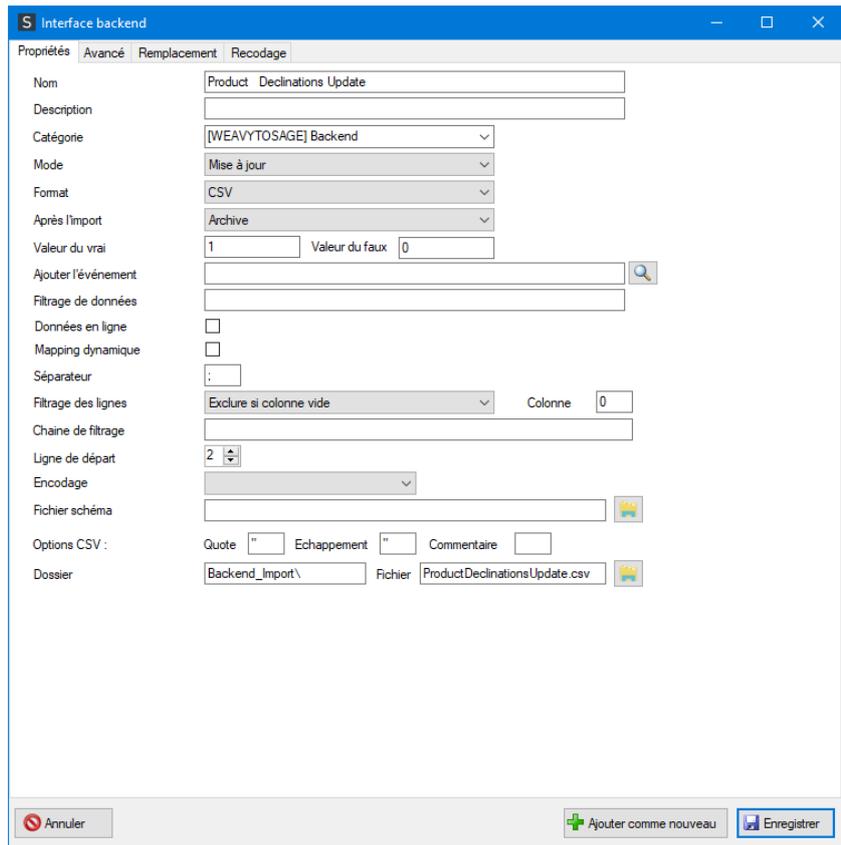


Mapping :

Product Declinations ProductDeclinations.csv		Product (sw_data_product)					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	CodeProduct	0	Product		Varchar (255)	CodeProduct	1
1	description	1	Description		Varchar (255)	description	2
2	productfamily_ID	2	Product / Family		ForeignKey	productfamily_ID	4
3	saleMultiple	3	Sale multiple		Double	saleMultiple	5
4	defaultPrice	4	Default price		Double	defaultPrice	6
5	purchasePrice	5	Purchase price		Double	purchasePrice	7
6	coefficient	6	Coefficient		Double	coefficient	9
7	weight	7	Weight		Double	weight	10
8	barcode	8	Barcode		Varchar (64)	barcode	14
9	packaging	9	Packaging		Varchar (64)	packaging	22
10	supplierReference	10	Supplier reference		Varchar (64)	supplierReference	32
11	supplierBarcode	11	Supplier barcode		Varchar (64)	supplierBarcode	33
12	blocked	12	Blocked		Bool	blocked	47
13	supplier_ID	13	Supplier		ForeignKey	supplier_ID	54
14	supplementaryDescription1	14	Supplementary description 1		Text (65535)	supplementaryDescription1	55
15	supplementaryDescription2	15	Supplementary description 2		Text (65535)	supplementaryDescription2	56
16	unused	16	Unused		Bool	unused	58
17	height	17	Height		Double	height	60
18	width	18	Width		Double	width	61
19	depth	19	Depth		Double	depth	62
20	productGeneric	20	Generic product		Bool	productGeneric	65
21	product_ID_parentDeclination	21	Parent product for declina...		ForeignKey	product_ID_parentDeclination	66
22	productdeclination_ID_level1	22	Variant / Level 1		ForeignKey	productdeclination_ID_level1	67
23	productdeclination_ID_level2	23	Variant / Level 2		ForeignKey	productdeclination_ID_level2	68
24	stockDeclination	24	Stock management with v...		Bool	stockDeclination	72
25	erpProductID	25	ERP Product ID		Varchar (255)	erpProductID	73
26	generictype_ID_weightunit	26	Weight unit		ForeignKey	generictype_ID_weightunit	13
27	generictype_ID_taxCode	27	Product tax code		ForeignKey	generictype_ID_taxCode	23
28	generictype_ID_sellingUnit	28	Selling unit		ForeignKey	generictype_ID_sellingUnit	57
29	generictype_ID_managementType	29	Management type		ForeignKey	generictype_ID_managementType	59
30	generictype_ID_lengthunit	30	Length unit		ForeignKey	generictype_ID_lengthunit	63
31	generictype_ID_brand	31	Brand		ForeignKey	generictype_ID_brand	64
32	final_w2s_FactPoids	32	final_w2s_FactPoids		Bool	final_w2s_FactPoids	75

p. Backend Product Declinations Update(sw_data_product)

Interface :

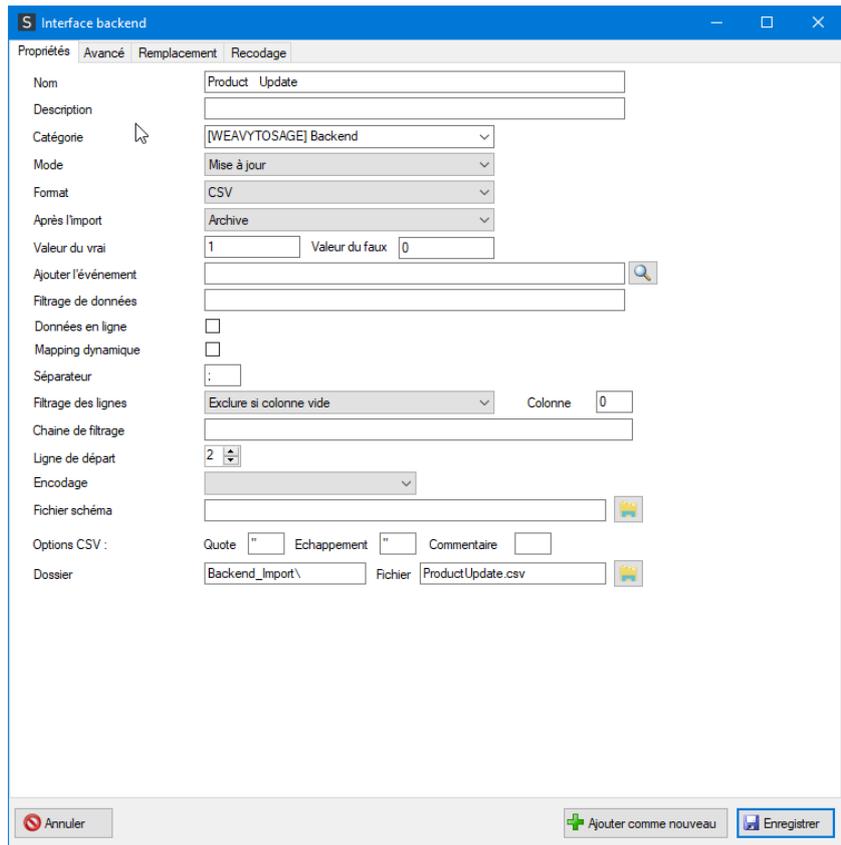


Mapping :

Product Declinations Update ProductDeclinationsUpdate.csv		Product (sw_data_product)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	CodeProduct	0	Product		Varchar (255)	CodeProduct	1
1	description	1	Description		Varchar (255)	description	2
2	productfamily_ID	2	Product / Family		ForeignKey	productfamily_ID	4
3	saleMultiple	3	Sale multiple		Double	saleMultiple	5
4	defaultPrice	4	Default price		Double	defaultPrice	6
5	purchasePrice	5	Purchase price		Double	purchasePrice	7
6	coefficient	6	Coefficient		Double	coefficient	9
7	weight	7	Weight		Double	weight	10
8	barcode	8	Barcode		Varchar (64)	barcode	14
9	packaging	9	Packaging		Varchar (64)	packaging	22
10	supplierReference	10	Supplier reference		Varchar (64)	supplierReference	32
11	supplierBarcode	11	Supplier barcode		Varchar (64)	supplierBarcode	33
12	blocked	12	Blocked		Bool	blocked	47
13	supplier_ID	13	Supplier		ForeignKey	supplier_ID	54
14	supplementaryDescription1	14	Supplementary description 1		Text (65535)	supplementaryDescription1	55
15	supplementaryDescription2	15	Supplementary description 2		Text (65535)	supplementaryDescription2	56
16	unused	16	Unused		Bool	unused	58
17	height	17	Height		Double	height	60
18	width	18	Width		Double	width	61
19	depth	19	Depth		Double	depth	62
20	productGeneric	20	Generic product		Bool	productGeneric	65
21	product_ID_parentDeclination	21	Parent product for declina...		ForeignKey	product_ID_parentDeclination	66
22	productdeclination_ID_level1	22	Variant / Level 1		ForeignKey	productdeclination_ID_level1	67
23	productdeclination_ID_level2	23	Variant / Level 2		ForeignKey	productdeclination_ID_level2	68
24	stockDeclination	24	Stock management with v...		Bool	stockDeclination	72
25	erpProductID	25	ERP Product ID		Varchar (255)	erpProductID	73
26	generictype_ID_weightunit	26	Weight unit		ForeignKey	generictype_ID_weightunit	13
27	generictype_ID_taxCode	27	Product tax code		ForeignKey	generictype_ID_taxCode	23
28	generictype_ID_sellingUnit	28	Selling unit		ForeignKey	generictype_ID_sellingUnit	57
29	generictype_ID_managementType	29	Management type		ForeignKey	generictype_ID_managementType	59
30	generictype_ID_lengthunit	30	Length unit		ForeignKey	generictype_ID_lengthunit	63
31	generictype_ID_brand	31	Brand		ForeignKey	generictype_ID_brand	64
32	final_w2s_FactPoids	32	final_w2s_FactPoids		Bool	final_w2s_FactPoids	75

q. Backend Product Update (sw_data_product)

Interface :



Mapping :

Product Update ProductUpdate.csv		Product (sw_data_product)					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	CodeProduct	0	Product		Varchar (255)	CodeProduct	1
1	description	1	Description		Varchar (255)	description	2
2	productfamily_ID	2	Product / Family		ForeignKey	productfamily_ID	4
3	saleMultiple	3	Sale multiple		Double	saleMultiple	5
4	defaultPrice	4	Default price		Double	defaultPrice	6
5	purchasePrice	5	Purchase price		Double	purchasePrice	7
6	coefficient	6	Coefficient		Double	coefficient	9
7	weight	7	Weight		Double	weight	10
8	barcode	8	Barcode		Varchar (64)	barcode	14
9	packaging	9	Packaging		Varchar (64)	packaging	22
10	supplierReference	10	Supplier reference		Varchar (64)	supplierReference	32
11	supplierBarcode	11	Supplier barcode		Varchar (64)	supplierBarcode	33
12	blocked	12	Blocked		Bool	blocked	47
13	supplier_ID	13	Supplier		ForeignKey	supplier_ID	54
14	supplementaryDescription1	14	Supplementary description 1		Text (65535)	supplementaryDescription1	55
15	supplementaryDescription2	15	Supplementary description 2		Text (65535)	supplementaryDescription2	56
16	unused	16	Unused		Bool	unused	58
17	height	17	Height		Double	height	60
18	width	18	Width		Double	width	61
19	depth	19	Depth		Double	depth	62
20	productGeneric	20	Generic product		Bool	productGeneric	65
21	product_ID_parentDeclination	21	Parent product for declina...		ForeignKey	product_ID_parentDeclination	66
22	productdeclination_ID_level1	22	Variant / Level 1		ForeignKey	productdeclination_ID_level1	67
23	productdeclination_ID_level2	23	Variant / Level 2		ForeignKey	productdeclination_ID_level2	68
24	stockDeclination	24	Stock management with v...		Bool	stockDeclination	72
25	erpProductID	25	ERP Product ID		Varchar (255)	erpProductID	73
26	generictype_ID_weightunit	26	Weight unit		ForeignKey	generictype_ID_weightunit	13
27	generictype_ID_taxCode	27	Product tax code		ForeignKey	generictype_ID_taxCode	23
28	generictype_ID_sellingUnit	28	Selling unit		ForeignKey	generictype_ID_sellingUnit	57
29	generictype_ID_managementType	29	Management type		ForeignKey	generictype_ID_managementType	59
30	generictype_ID_lengthunit	30	Length unit		ForeignKey	generictype_ID_lengthunit	63
31	generictype_ID_brand	31	Brand		ForeignKey	generictype_ID_brand	64
32	final_w2s_FactPoids	32	final_w2s_FactPoids		Bool	final_w2s_FactPoids	75

r. Backend Product / Conditionnement (sw_data_productdeclination)

Interface :

Mapping :

Product / Conditionnement Product_conditionnement.csv		Product / Variant (sw_data_productdeclination)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Dictio...
0	productdeclinationtype_ID	0	Product / Variant / Type		ForeignKey	productdeclinationtype_ID	2
1	label	1	Label		Varchar (255)	label	3
2	orderNumber	2	Order number		Int	orderNumber	7
3	codeproductdeclination	3	Product / Variant		Varchar (255)	codeproductdeclination	1

- s. Backend Product / Conditionnement / Type (sw_data_productdeclinationtype)

Interface :

Mapping :

Product / Conditionnements / Type Product / Variant / Type (sw_data_productdeclinationtype)
[Product_Conditionnement_Type.c](#)

Auto mapping

Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	codeproductdeclinationtype	0	Product / Variant / Type		Varchar (255)	codeproductdeclinationtype	1
1	label	1	Label		Varchar (255)	label	2

t. Backend Product / Declination (sw_data_productdeclination)

Interface :

Mapping :

Product / Declination Product_Declination.csv		Article / Déclinaison (sw_data_productdeclination)					
Pos.	Champ	Pos.	Champ	Propriétés	Type	Nom	Diction...
0	productdeclinationtype_ID	0	Article / Déclinaison / Type		ForeignKey	productdeclinationtype_ID	2
1	label	1	Libellé		Varchar (255)	label	3
2	orderNumber	2	Numéro d'ordre		Int	orderNumber	7
3	codeproductdeclination	3	Article / Déclinaison		Varchar (255)	codeproductdeclination	1

u. Backend Product / Declination / Type (sw_data_productdeclinationtype)

Interface :

Mapping :

Product / Declination / Type Product_Declination_Type.csv		Article / Déclinaison / Type (sw_data_productdeclinationtype)					
Pos.	Champ	Pos.	Champ	Propriétés	Type	Nom	Diction...
0	codeproductdeclinationtype	0	Article / Déclinaison / Type		Varchar (255)	codeproductdeclinationtype	1
1	label	1	Libellé		Varchar (255)	label	2

v. Backend Product / Depot (sw_data_productdepot)

Interface :

Mapping :

Product / Depot		Article / Dépôt (sw_data_productdepot)					
Pos.	Champ	Pos.	Champ	Propriétés	Type	Nom	Diction...
0	CodeProductdepot	0	Article / Dépôt		Varchar (255)	CodeProductdepot	1
1	label	1	Libellé		Varchar (255)	label	2
2	main	2	Principal		Bool	main	4

w. Backend Product / Family (sw_data_productfamily)

Interface :

Mapping :

Product / Family Product_Family.csv		Article / Famille (sw_data_productfamily)					
Pos.	Champ	Pos.	Champ	Propriétés	Type	Nom	Diction...
0	CodeProductFamily	0	Article / Famille		Varchar (255)	CodeProductfamily	1
1	label	1	Libellé		Varchar (255)	label	2
2	productfamily_ID_parentFamily	2	Famille parent		ForeignKey	productfamily_ID_parentFamily	4
3	hierarchicalLevel	3	Niveau hiérarchique		Int	hierarchicalLevel	3

- x. Backend Product / Image (sw_data_productpicture)
Le transfert d'images qui sera réalisé dans le [transfert d'image](#) du ControlCenter va vous générer un dossier Media qui sera créé dans l'arborescence avec un dossier par article.

Le dossier qui contiendra ce dossier Media est défini au moment du paramétrage du [transfert de fichier](#).

Ensuite on va donc paramétrer une interface backend de la manière suivante.

Interface :

Mapping :

Product / Image		Article / Image (sw_data_productpicture)					
Pos.	Champ	Pos.	Champ	Propriétés	Type	Nom	Diction...
0	File name	1	Article / Image		Varchar (255)	CodeProductpicture	1
1	File name without extension	1	Article		ForeignKey	product_ID	3
2	Extension	6	Données binaires		MediumBlob	binaryData	2
3	Length		Image par défaut		Bool	defaultPicture	4
4	Last write time		Nom du fichier		Varchar (255)	fileName	5
5	Creation time						
6	Binary datas						

y. Backend Product / Stock (sw_data_productstock)

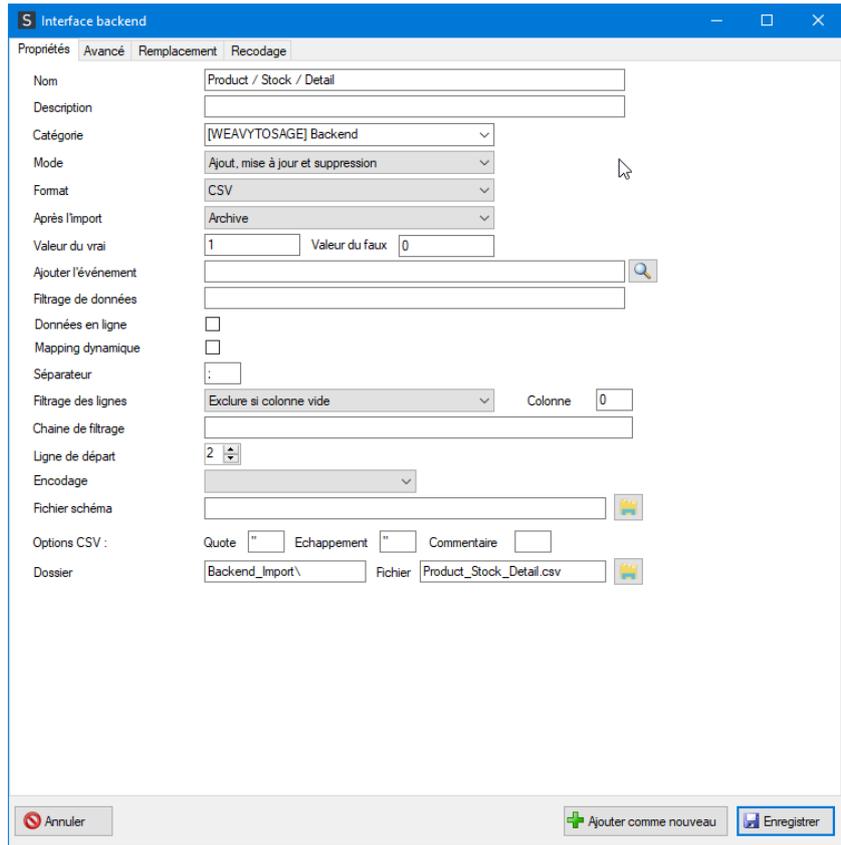
Interface :

Mapping :

Product / Stock		Product / Stock (sw_data_productstock)					
Pos.	Field	Pos.	Field	Properties	Type	Name	Diction...
0	product_ID	0	Product		ForeignKey	product_ID	2
1	productdepot_ID	1	Product / Depot		ForeignKey	productdepot_ID	11
2	minQuantity	2	Minimum qty		Double	minQuantity	5
3	maxQuantity	3	Maximum qty		Double	maxQuantity	6
4	storedQuantity	4	Stored qty		Double	storedQuantity	7
5	reservedQuantity	5	Reserved qty		Double	reservedQuantity	8
6	orderQuantity	6	Ordered qty		Double	orderQuantity	9
7	mainStorage	7	Main storage		Bool	mainStorage	10
8	CodeProductstock	8	Product / Stock		Varchar (255)	CodeProductstock	1

z. Backend Product / Stock / Detail (sw_data_productstockdetail)

Interface :



Mapping :

Product / Stock / Detail Product_Stock_Detail.csv		Product / Stock / Detail (sw_data_productstockdetail)					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	batchSerialNumber	0	Batch/serial number		Varchar (255)	batchSerialNumber	3
1	productstockqualification_ID	1	Product / Stock / Qualifica...		ForeignKey	productstockqualification_ID	4
2	quantity	2	Quantity		Double	quantity	5
3	shelfLife	3	Shelf life		DateTime	shelfLife	6
4	codeproductstockdetail	4	Product / Stock / Detail		Varchar (255)	codeproductstockdetail	1
5	productstock_ID	5	Product / Stock		ForeignKey	productstock_ID	2

aa. Backend SalesCondition (sw_data_salescondition)

Interface :

Mapping :

SalesCondition		Sales condition (sw_data_salescondition)					
SalesCondition.csv		Auto mapping					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	codeSalescondition	0	Salescondition		Varchar (255)	CodeSalescondition	1
1	customer_ID	1	Customer		ForeignKey	customer_ID	2
2	product_ID	2	Product		ForeignKey	product_ID	7
3	productfamily_ID	3	Product / Family		ForeignKey	productfamily_ID	13
4	priceCategory_ID	4	Price Category		ForeignKey	pricecategory_ID	14
5	currency_ID	5	Currency		ForeignKey	currency_ID	18
6	minQuantity	6	Minimum quantity		Double	minQuantity	19
7	maxQuantity	7	Maximum quantity		Double	maxQuantity	20
8	turnoverMin	8	Turnover Min		Double	turnoverMin	21
9	turnoverMax	9	Turnover Max		Double	turnoverMax	22
10	startDate	10	Start date		Timestamp	startDate	23
11	endDate	11	End date		Timestamp	endDate	24
12	netPrice	12	Net price		Double	netPrice	25
13	discountValue1	13	Discount 1		Double	discountValue1	28
14	discountType1	14	Discount type 1		Int	discountType1	29
15	discountValue2	15	Discount 2		Double	discountValue2	30
16	discountType2	16	Discount type 2		Int	discountType2	31
17	discountValue3	17	Discount 3		Double	discountValue3	32
18	discountType3	18	Discount type 3		Int	discountType3	33
19	coefficient	19	Coefficient		Double	coefficient	34
20	currencyPrice	20	Price (currency)		Double	currencyPrice	40
21	level	21	Level		Int	level	41
22	global	22	Global		Bool	global	42

bb. Backend Statistic (sw_data_statistic)

Interface :

Mapping :

Statistic		Statistic (sw_data_statistic)					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	year	0	Year		Int	year	2
1	month	1	Month		Int	month	3
2	customer_ID	2	Customer		ForeignKey	customer_ID	4
3	baseuser_ID	3	Base / User		ForeignKey	baseuser_ID	5
4	product_ID	4	Product		ForeignKey	product_ID	6
5	quantity	5	Quantity		Double	quantity	7
6	amount	6	Amount		Double	amount	8
7	margin	7	Margin		Double	margin	9
8	CodeStatistic	8	Statistic		Varchar (255)	CodeStatistic	1

cc. Backend Supplier (sw_data_supplier)

Interface :

Mapping :

Supplier Supplier.csv		Fournisseurs (sw_data_supplier)					
		* Auto mapping					
Pos.	Champ	Pos.	Champ	Propriétés	Type	Nom	Diction...
0	CodeSupplier	0	Fournisseurs		Varchar (255)	CodeSupplier	1
1	label	1	Libellé		Varchar (64)	label	2
2	phoneNumber	2	Téléphone		Varchar (64)	phoneNumber	3
3	postalCode	3	Code postal		Varchar (64)	postalCode	4
4	city	4	Ville		Varchar (64)	city	5

dd. Backend Tax (sw_data_tax)

Interface :

S Interface backend

Propriétés | Avancé | Remplacement | Recodage

Nom: Tax

Description: Tarif Taxes

Catégorie: [WEAVYTOSAGE] Backend

Mode: Ajout et mise à jour

Format: CSV

Après l'import: Archive

Valeur du vrai: 1 Valeur du faux: 0

Ajouter l'événement: []

Filtrage de données: []

Données en ligne:

Mapping dynamique:

Séparateur: :

Filtrage des lignes: Exclure si colonne vide Colonne: 0

Chaîne de filtrage: []

Ligne de départ: 2

Encodage: []

Fichier schéma: []

Options CSV: Quote " Echappement " Commentaire []

Dossier: Backend_import\ Fichier: Tax.csv

Annuler Ajouter comme nouveau Enregistrer

Mapping :

Tax		Tax (sw_data_tax)					
Tax.csv		Auto mapping					
Pos.	Field	Pos.	Field	Propertes	Type	Name	Diction...
0	taxRate	0	Tax rate		Double	taxRate	4
1	description	1	Description		Varchar (64)	description	7
2	taxPriority	2	taxPriority		Int	taxPriority	6
3	generictype_ID_customerTax	3	Customer tax code		ForeignKey	generictype_ID_customerTax	3
4	generictype_ID_productTax	4	Product tax code		ForeignKey	generictype_ID_productTax	2
5	CodeTax	5	Tax		Varchar (255)	CodeTax	1

ee. Planification

S Scheduler

Name [WeavyToSage] Import

Description

Add event

Folder condition

File condition

File end

Mail

Days of the week

Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

Select all

Days of the month

1
2
3
4
5
6
7
8
9
10
11
12
13

Select all

Month

January
February
March
April
May
June
July
August
September
October
November
December

Select all

Interval of execution

Start 00 : 00

End 00 : 00

Duration 1 min

Fixed schedule

New schedule 00 : 00

Cancel Apply

Saisir un nom pour la planification et choisir les backends indiqués dans l'ordre ci-dessous.

Name	Mode	Post import
Generic Type	AddUpdate	Archive
Currency	AddUpdate	Archive
BaseUser	AddUpdate	Archive
Supplier	AddUpdate	Archive
PriceCategory	AddUpdate	Archive
Customer	AddUpdate	Archive
Customer / Address	AddUpdate	Archive
Customer / Contact	AddUpdate	Archive
Customer / Credit	AddUpdate	Archive
Customer / User	AddUpdate	Archive
Product / Family	AddUpdate	Archive
Product / Depot	AddUpdate	Archive
Product / Declination / Type	AddUpdate	Archive
Product / Declination	AddUpdate	Archive
Product / Conditionnements / Type	AddUpdate	Archive
Product / Conditionnement	AddUpdate	Archive
Product	AddUpdate	Archive
Product Declinations	AddUpdate	Archive
Product Declinations Update	Update	Archive
Product Conditionnements	AddUpdate	Archive
Product Conditionnements Update	Update	Archive
Product Update	Update	Archive
Product / Image	AddUpdate	Delete
Product / Stock	AddUpdateDelete	Archive
Product / Stock / Detail	AddUpdateDelete	Archive
Histo / Header	AddUpdate	Archive
Histo / Detail	AddUpdate	Archive
SalesCondition	AddUpdateDelete	Archive
Statistic	AddUpdate	Archive
Tax	AddUpdate	Archive

En bas à gauche, vous trouverez l'ID qu'il faudra aller remplir dans le [Control Center](#) pour que le traitement se lance automatiquement à la fin du transfert de fichier.

Name **[WeavyToSage] Import - ID 64 - description - Mail**

3. Control Center

I. Téléchargement / Installation

Aller sur <https://apps.weavy.divalto.com/> et connecter vous avec les identifiants fournis par Divalto pour les partenaires.



Cliquer sur .

Cliquer ensuite sur la version « release ».

Puis sur télécharger.

Aller dans les propriétés du fichier télécharger et il faudra cocher cette partie et valider.

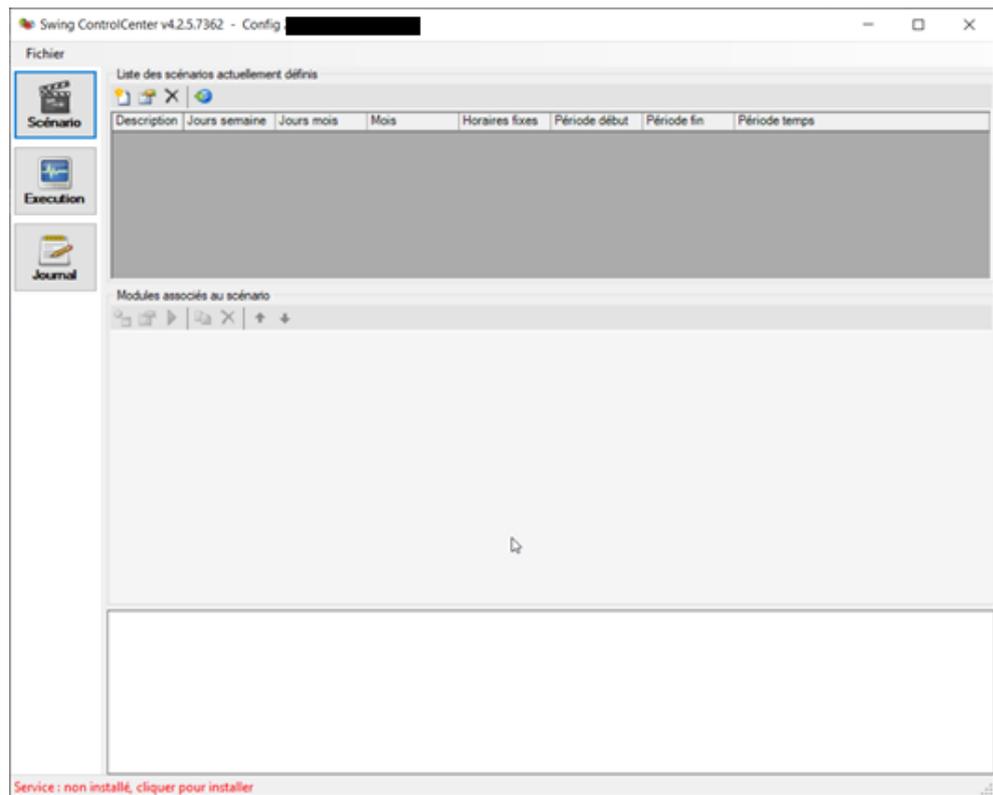
Sécurité : Ce fichier provient d'un autre ordinateur et peut éventuellement être bloqué pour protéger cet ordinateur. Débloquer

Dézipper le fichier téléchargé, un dossier « SwingControlCenter » est créé

Dans ce dossier lancer « SwingControlCenter.exe » en mode administrateur, penser à toujours lancer le programme en mode administrateur.

Dans l'écran suivant il faudra renseigner votre code société fourni par Divalto et un mot de passe pour l'accès distant.

Ensuite on arrive sur l'écran principal du Control Center :



En bas on remarque un **Service : non installé, cliquer pour installer**.

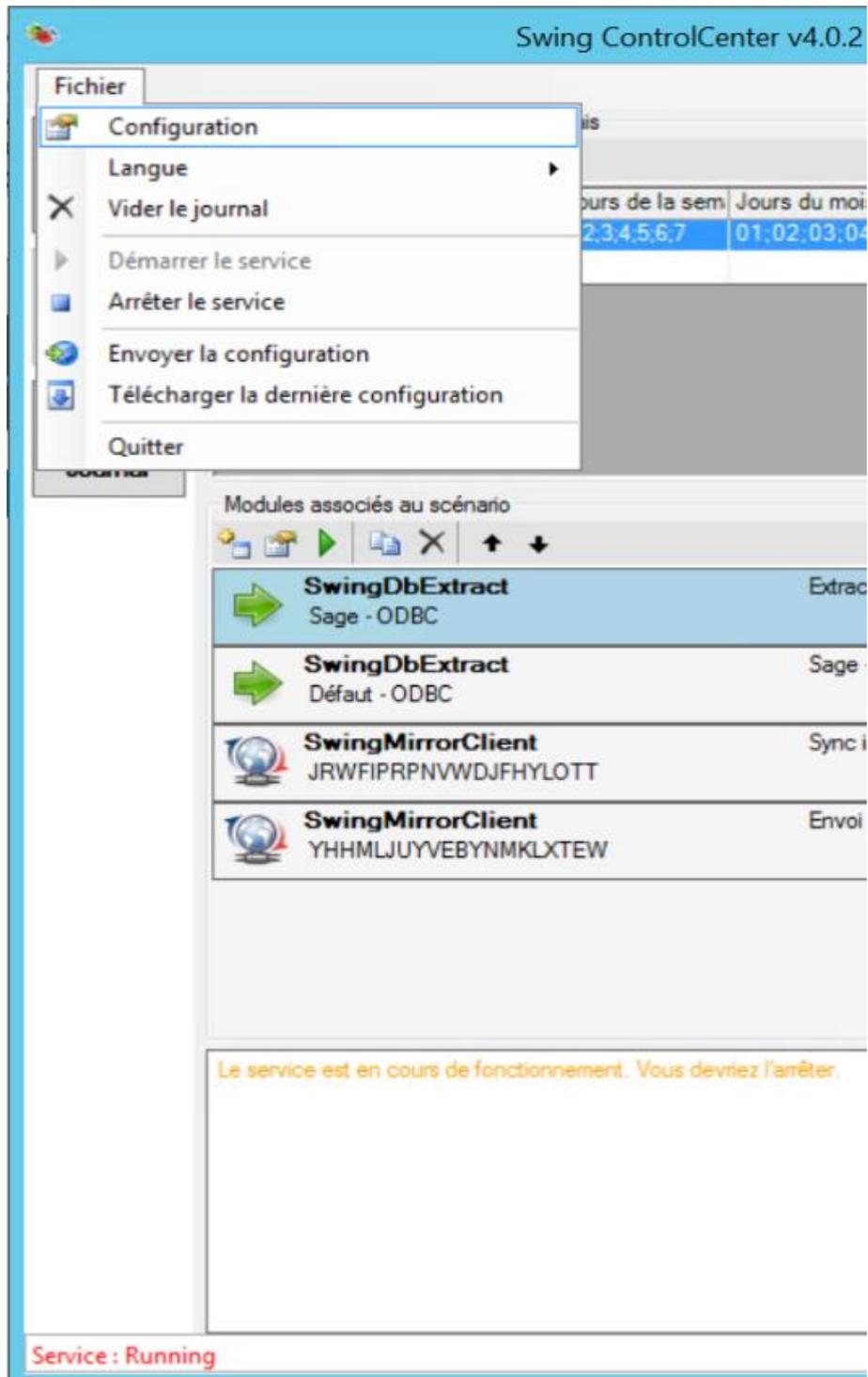
Cliquer dessus pour installer le service qui permettra une exécution en tâche de fond via le service Windows dédié.

S'il s'est bien installé vous devez avoir **Service : Stopped** à la place.

II. Paramétrage

Lancer le Control Center en mode Administrateur.

Si le service est lancé :



Fichier -> Arrêter le service.

Ensuite aller dans Fichier -> Configuration et renseigner le code société de votre projet actuel, puis tabuler. Un  indique qu'il est correct et demande un mot de

passer pour la sauvegarde et la récupération des configurations sur le cloud Weavy par la suite. Il ne reste plus qu'à valider.

III. Scénario

Maintenant il faut ajouter un scénario qui va nous permettre d'extraire les données de l'ERP Sage dont on a besoin.

Note : Pour ceux qui partent d'une installation Weavy vierge, vous pouvez récupérer le dossier [ConfigMain](#) dans le dossier ControlCenter du Package et le copier dans le dossier du ControlCenter.

Pour ce faire cliquer sur . Une nouvelle fenêtre s'ouvre :

Configuration d'un scénario

Description du scénario

Définitions des jours d'activité

Jours de la semaine

Jours du mois

Mois de l'année

Périodicité des exécutions

Horaires fixes d'exécution

Sur cette fenêtre vous pouvez définir la récurrence de lancement du scénario.

IV. Module

A chaque scénario on peut associer plusieurs modules, les modules permettent d'effectuer des actions spécifiques (extraction de données, transfert de fichier, etc.)

Liste des scénarios actuellement définis

Description	Jours de la sem	Jours du mois	Mois	Hoaires
Sage / Kit de Connexion Export	1,2,3,4,5,6,7	01,02,03,04,05,06,07,08,09,10,11	01,02,03,04,05,06,07,08,09,10,11,12	
Sage / Kit de Connexion Import				

Modules associés au scénario

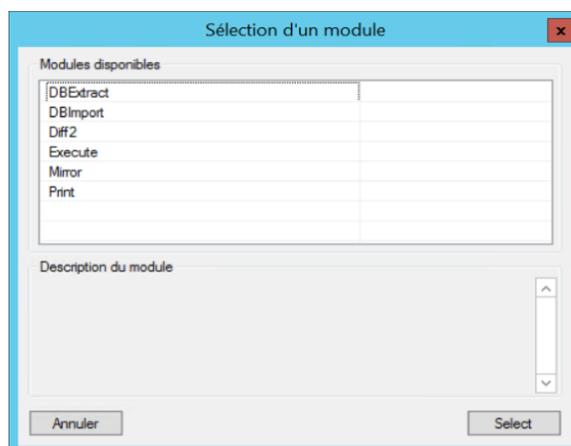
	SwingDbExtract Sage - ODBC	Extraction des données Sage
	SwingDbExtract Défaut - ODBC	Sage - Extraction des médias
	SwingMirrorClient JRWFIPRPNVWDJFHLYOTT	Sync images
	SwingMirrorClient YHHMLJUYVEBYNMKLXTEW	Envoi des données

Dans notre cas nous auront besoin d'un module pour l'extraction de données et d'un second pour le transfert des médias, en l'occurrence ici les photos des articles.

- a. Module d'extraction de données (Extraction csv des données de SAGE)

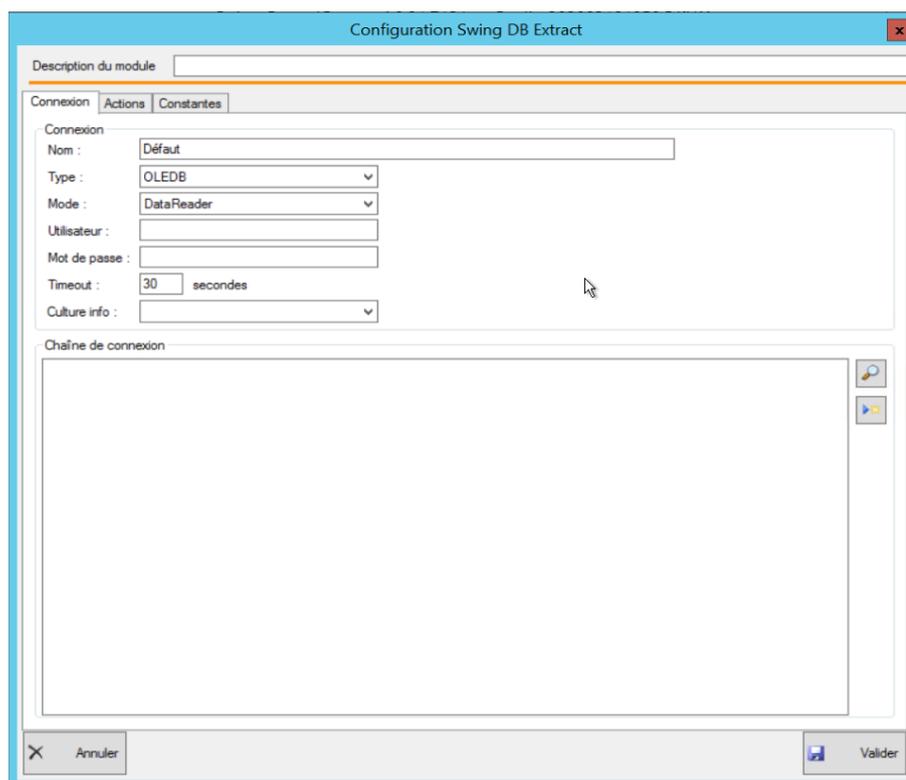
Pour la création du premier module, cliquer sur .

On vous propose alors de choisir le type de module :



Choisir « DBExtract ».

Une nouvelle fenêtre s'ouvre (le champ description du module s'affiche sur l'écran de base pour s'y retrouver plus simplement) :



Onglet « Connexion »

Plusieurs choses à remplir ici :

Un Nom : il apparaîtra sur la fenêtre de base, vous pouvez mettre le même nom que la connexion ODBC qui sera utilisé (voir un peu plus bas).

Un Type : ici on utilisera une connexion « ODBC ».

Un Mode : on laissera en « DataReader ».

Utilisateur et Mot de passe ne seront pas remplis car le Programme tournera via un service windows en tâche de fond.

Un Timeout : A définir selon la quantité de données à récupérer.

Un champ Culture info : Il faudra renseigner la langue utilisée.

Et pour finir la chaîne de connexion utilisée que vous pourrez choisir via le bouton . Une liste s'ouvrira où il faudra choisir la connexion souhaitée.

Le bouton  permet de vérifier que la connexion choisie fonctionne correctement.

Pour créer la connexion ODBC, faites une recherche de « ODBC » dans le menu démarrer de l'ordinateur et vous trouverez ces deux résultats :



Nous allons détailler la procédure à faire sur l'une des deux interfaces de paramétrage de la Source de données ODBC (version 32 bit), il faudra la reproduire sur l'autre (version 64 bit). Attention, les deux versions doivent être configurées à l'identique :

Lancer l'application ODBC Data Sources (32-bit) (ou Sources de Données ODBC) en administrateur avec un clic droit.

Aller dans l'onglet « Source de données système » ou « DSN System ».

Cliquer sur le bouton « Ajouter ».

Dans la nouvelle fenêtre, choisir « SQL server native client 11.0 » (dépend de la version de votre SQL Server installée, la version peut différer) puis cliquer sur « Terminer ».

Dans « Nom », définir un nom simple qui sera repris dans la liste des connexions disponibles.

Choisir le serveur où se trouve la base et cliquer sur « Suivant ».

Cocher l'option d'authentification « Avec l'authentification Windows intégré » et cliquer sur « Suivant ».

Cocher « Changer la base de données par défaut » et choisir la base de données à utiliser puis faire « Suivant ».

Sur le dernier écran, faire « Terminer »

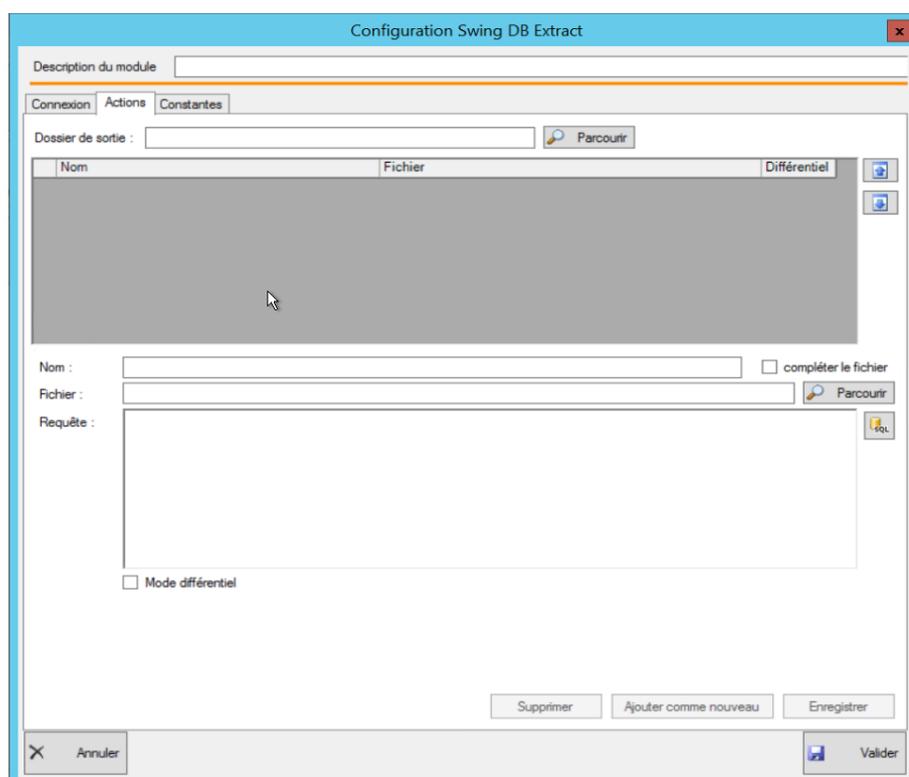
Il vous affiche ensuite un écran récapitulatif du paramétrage, cliquer sur « Tester la source de données ».

Si tout est OK le message « TESTS EFFECTUÉS AVEC SUCCÈS » apparaît, sinon il vous faudra rectifier votre paramétrage. Vous pouvez faire « Ok » et encore « Ok » pour terminer le paramétrage.

Il ne vous reste plus qu'à reproduire ce paramétrage sur l'autre version de gestionnaire des Sources de données ODBC.

Onglet « Action »

Dans cet onglet on va indiquer les différentes extractions qui seront à réaliser dans la base.



Dans un premier temps il va falloir définir un dossier d'export pour les fichiers qui seront générés depuis ce module.

Un nom devra être défini pour chaque action générée ainsi qu'un fichier CSV.

Il faudra indiquer une requête SELECT qui générera le fichier CSV avec les colonnes spécifiées comme champs dans la requête.

Le bouton  permet d'ouvrir un requêteur pour tester la requête.

Voici les liens vers les différents scripts SQL à utiliser dans le Control Center :

[BaseUser](#)

[Currency](#)

[Customer](#)

[Customer_Address](#)

[Customer_Contact](#)

[Customer_Credit](#)

[Customer_User](#)

[GenericTypes](#)

[Histo_Detail](#)

[Histo_Header](#)

[PriceCategory](#)

[Product](#)

[Product_Declination](#)

[Product_Declination_Type](#)

[Product_Depot](#)

[Product_Family](#)

[Product_Stock](#)

[Product_Stock_Detail](#)

[ProductDeclinations](#)

[ProductDeclinationsUpdate](#)

[ProductUpdate](#)

[SalesCondition](#)

[Statistic](#)

[Supplier](#)

[Tax](#)

Onglet « Constantes »

Pour la constante « COLLATE », il faut indiquer le collate de la base de données correspondant à la connexion ODBC choisi.

Constante :

Valeur :

Pour « ENVIRONMENTID », se référer à la partie [WeavyToSage](#) plus bas sur les tâches pour trouver id de l'environnement (pour le premier lancement on peut laisser 1)

Constante :

Valeur :

Pour « BDD », il faut renseigner la base de données créée par WeavyToSage

Constante :

Valeur :

Pour « SEPARATOR », il faut éviter de prendre un caractère qui est utilisé dans les références de SAGE (Si « _ » est utilisé dans les références vous pouvez par exemples utiliser « | ». Il faudra renseigner ce même caractère dans la partie [TaskIntervention](#) et [TaskSalesdocuments](#) plus bas.)

Constante :

Valeur :

Constante :

Valeur :

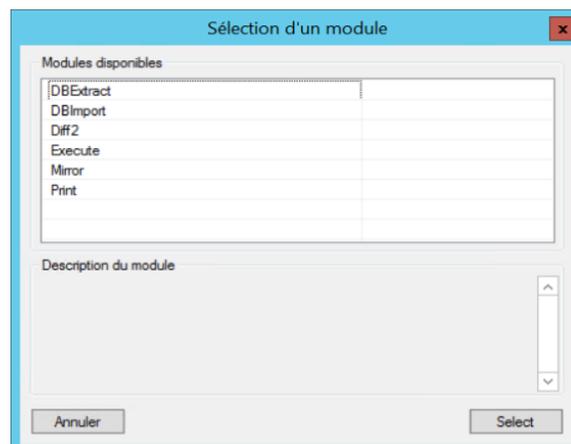
Constante :

Valeur :

- b. Module d'extraction des images articles (Extraction csv des chemins d'images SAGE)

Pour la création du premier module, cliquer sur .

On vous propose alors de choisir le type de module :



Nous choisirons « DBExtract ».

La fenêtre suivante s'affiche (le champ description du module s'affiche sur l'écran de base pour s'y retrouver plus simplement) :

Onglet « Connexion »

Plusieurs choses à remplir dans cet écran :

Un Nom : il apparaîtra sur la fenêtre de base, vous pouvez mettre le même nom que la connexion ODBC qui sera utilisée (voir un peu plus bas).

Un Type : ici on utilisera une connexion « ODBC ».

Un Mode : on laissera en « DataReader ».

Utilisateur et Mot de passe ne seront pas remplis car le Programme tournera via un service.

Un Timeout : A définir selon la quantité de données à récupérer.

Un champ Culture info : Il faudra renseigner la langue utilisée.

Et pour finir la chaîne de connexion utilisée que vous pourrez choisir via le

bouton . Une liste s'ouvrira dans laquelle il faudra choisir la

connexion. Le bouton  permet de vérifier que la connexion choisit fonctionne correctement.

Pour créer la connexion ODBC, faites une recherche de « ODBC » dans le menu démarrer vous trouverez ces deux résultats :



Nous allons détailler la procédure à faire sur une des deux versions (par exemple 32-bit), puis il faudra la reproduire sur l'autre version (64-bit) :

Lancer l'application en administrateur avec un clic droit.

Aller dans l'onglet « Source de données système » ou « DSN System ».

Cliquer sur le bouton « Ajouter ».

Dans la nouvelle fenêtre, choisir « SQL server native client 11.0 » puis cliquer sur « Terminer ».

Dans « Nom », définir un nom simple qui sera repris dans la liste des connexions disponibles.

Choisir le serveur où se trouve la base et cliquer sur « Suivant ».

Cocher l'option « Avec l'authentification Windows intégré » et cliquer sur « Suivant ».

Cocher « Changer la base de données par défaut » et choisir la base de données à utiliser puis cliquer sur « Suivant ».

Sur le dernier écran, cliquer sur « Terminer »

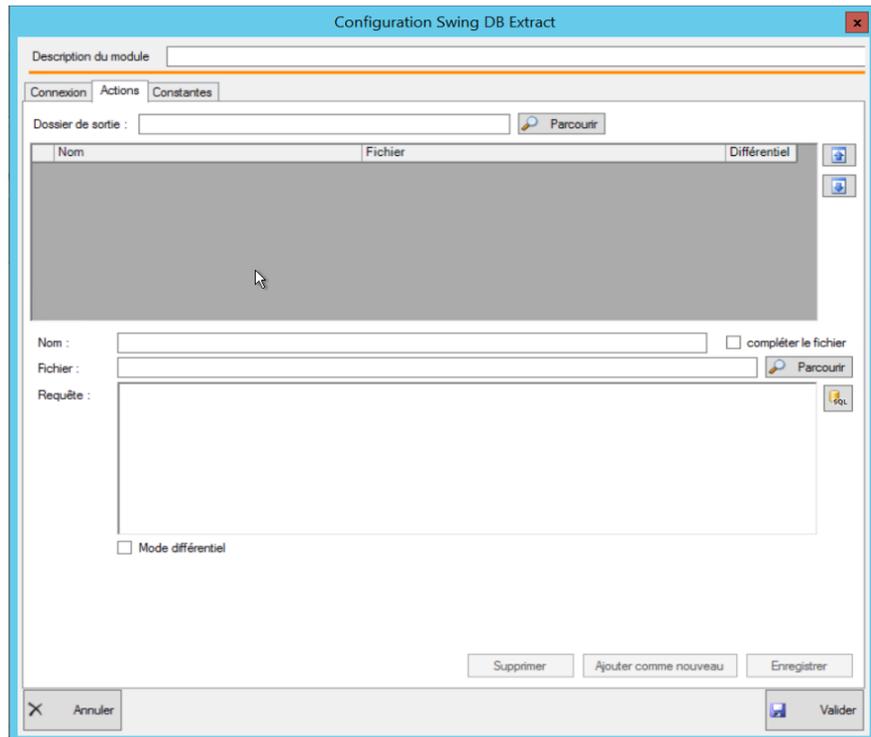
Un écran récapitulatif du paramétrage s'affiche. Cliquer sur « Tester la source de données ».

Si tout est OK le message « TESTS EFFECTUÉS AVEC SUCCÈS » s'affiche. Sinon il vous faudra rectifier votre paramétrage. Vous pouvez cliquer sur « Ok » pour valider.

Il ne vous reste plus qu'à reproduire ce paramétrage sur l'autre version du programme de configuration des sources de données ODBC.

Onglet « Action »

Dans cet onglet on va indiquer les différentes extractions qui seront à réaliser dans la base.



Dans un premier temps il va falloir définir un dossier d'export pour les fichiers qui seront générés depuis ce module. (Différent de l'autre dossier d'export)

Un nom devra être défini pour chaque action générée ainsi qu'un fichier CSV.

Il faudra indiquer une requête SELECT qui générera le fichier avec les colonnes correspondant aux champs de la requête.

Le bouton  permet d'ouvrir un requêteur pour tester la requête.

Voici les liens vers les différents scripts SQL à utiliser dans ce module ControlCenter :

[Media](#)

Onglet « Constantes »

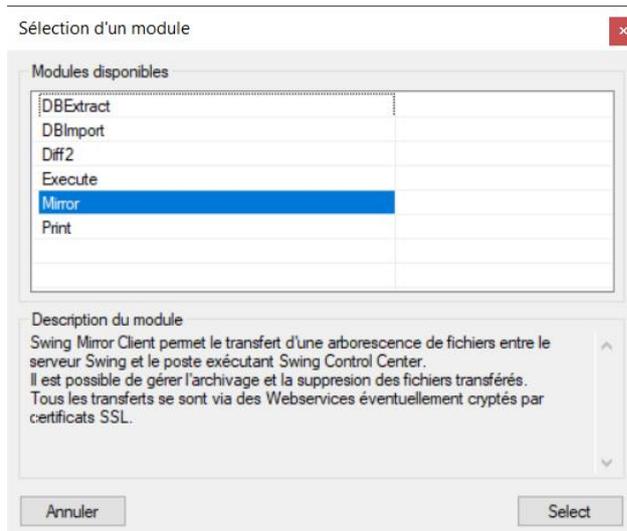
Ajouter les constantes suivantes :

Constante :
 Valeur :

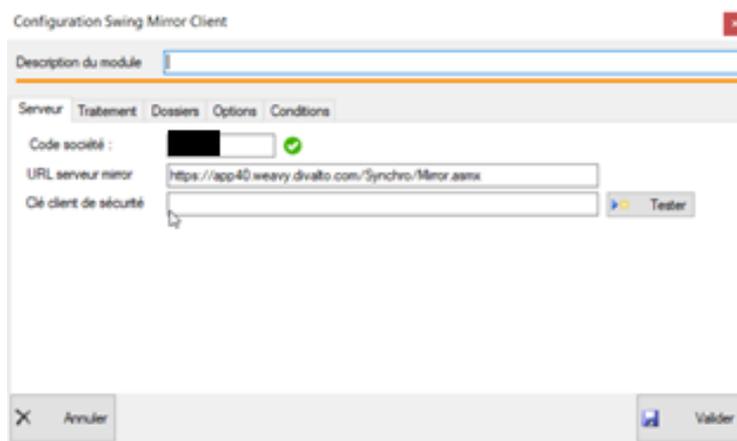
Il faudra remplacer la valeur « Votre_chemin » par le chemin où sont stockés vos images d'articles Sage. En principe il s'agit du dossier Multimedia de votre installation Sage 100 cloud (se référer à la documentation Sage 100 cloud).

c. [Module de transfert des images articles \(Envoie des images\)](#)

Pour ce module on va choisir l'option « Mirror » :



La fenêtre suivante apparaît :



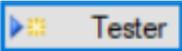
Onglet « Serveur »

Pour effectuer des transferts de fichiers depuis ou vers le serveur Weavy il vous faudra récupérer une clé de sécurité.

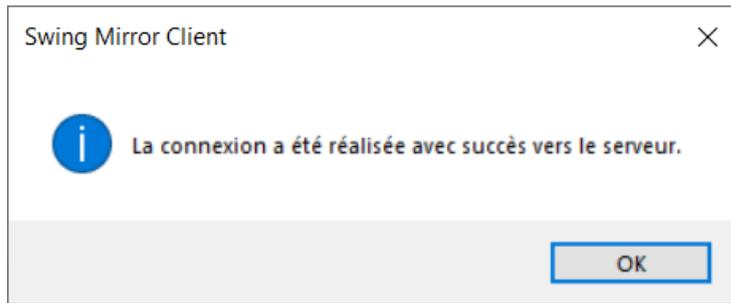
Pour ce faire on a généré une clé de sécurité pour récupérer les images dans le paramétrage du Studio : voir « [Document\ProductImage](#) ».

Remplir le code société Weavy du projet dans « Code société ».

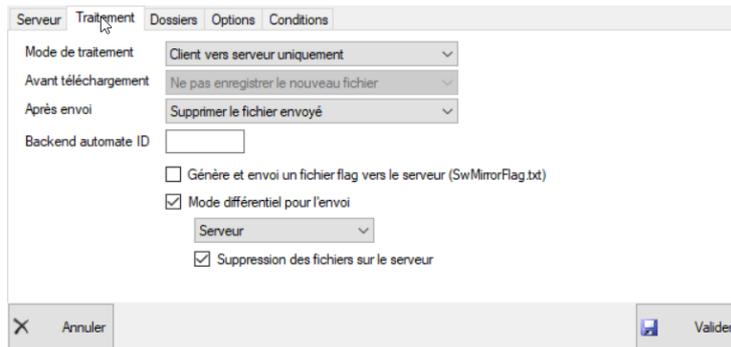
Coller la clé de sécurité précédente dans le champ « Clé client de sécurité ».

Appuyer sur  pour valider le bon fonctionnement.

Vous obtenez ce message :

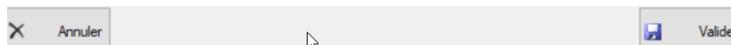


Onglet « Traitement »



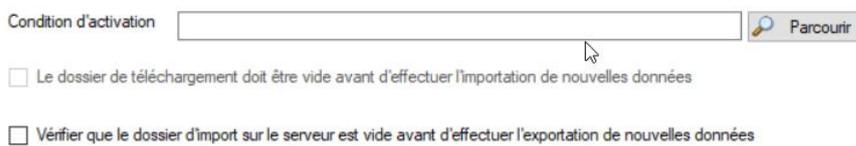
Onglet « Dossier »

Ici on va indiquer où on a enregistré nos fichiers du module précédent.



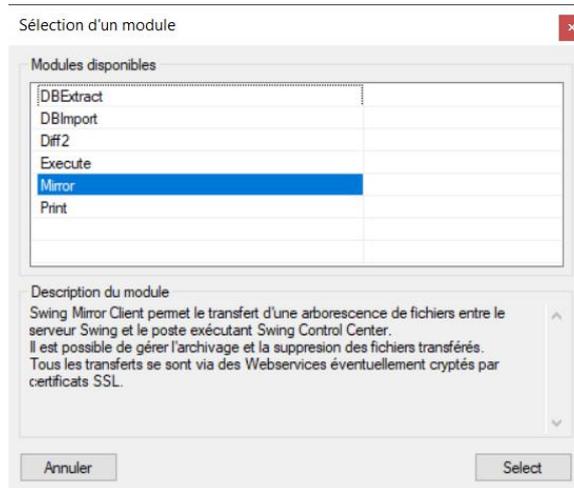
Remplacer « ... » par le chemin d'export du fichier d'images.

Onglet « Conditions »

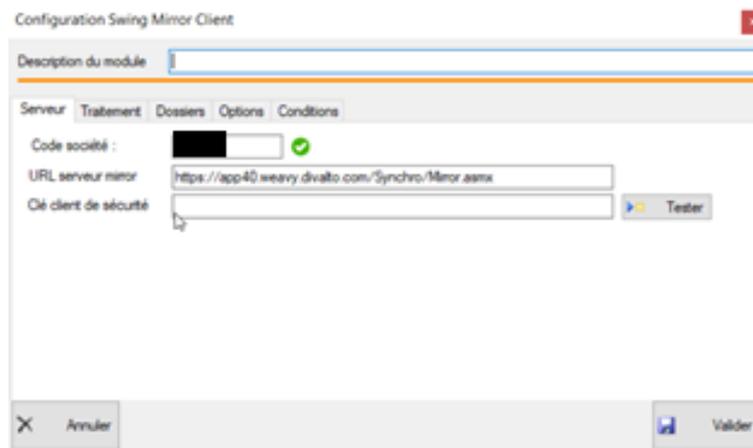


Cocher « Vérifier que le dossier d'import soit vide ».

- d. Module de transfert de données (Envoi des csv)
Pour ce module on va choisir l'option « Mirror » :



La fenêtre suivante apparaît :



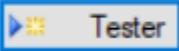
Onglet « Serveur »

Pour effectuer des transferts de fichiers depuis ou vers le serveur Weavy il vous faudra récupérer une clé de sécurité.

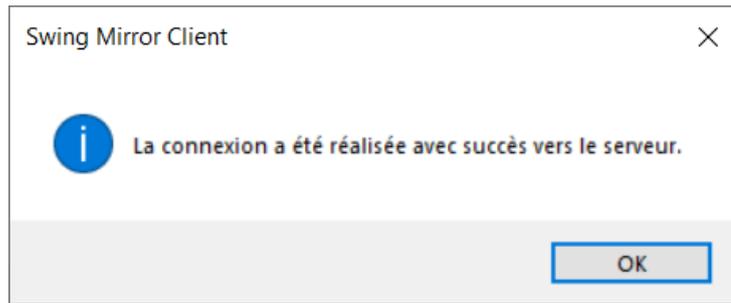
Pour ce faire, on a généré une clé de sécurité pour récupérer les csv lors du paramétrage du Studio : voir « [Backend Import](#) ».

Remplir le code société Weavy du projet dans « Code société ».

Coller la clé copiée à l'étape précédente dans le champ « Clé client de sécurité ».

Vous pouvez appuyer sur  pour voir si vous avez généré la clé correctement.

Si la clé est fonctionnelle vous aurez ce message :



Onglet « Traitement »

Mode de traitement

Avant téléchargement

Après envoi

Backend automate ID

Génère et envoi un fichier flag vers le serveur (SwMimorFlag.txt)

Mode différentiel pour l'envoi

Suppression des fichiers sur le serveur

Voici le paramétrage de l'onglet Traitement. On remplira le champ Backend automate ID par l'ID généré plus tôt dans le [studio](#) pour exécuter automatiquement le traitement d'import qui doit clôturer l'import de ces fichiers côté Cloud Weavy.

Note : Au premier lancement, si vous voulez vérifier le mapping dans le studio ne remplissez pas ce champ pour la première exécution.

Onglet « Dossier »

Ici on va indiquer où l'on a enregistré nos fichiers du module précédent.

On mettra en filtre « *.csv ».

Dossier de téléchargement

Source d'envoi

Filtres d'envoi (masques) :

Onglet « Conditions »

Condition d'activation

Le dossier de téléchargement doit être vide avant d'effectuer l'importation de nouvelles données

Vérifier que le dossier d'import sur le serveur est vide avant d'effectuer l'exportation de nouvelles données

Cocher « Vérifier que le dossier d'import soit vide ».

V. Lancement du Control Center

Si vous avez effectué la méthode simple avec l'import directement des backend contenus dans le package, repris directement le dossier configMain du ControlCenter et vérifié le paramétrage du scénario et des modules : Vous pouvez directement lancer le service.

Note : si vous voulez vérifier les mapping des csv par rapport aux backend, il faut enlever le « Backend automate ID » du dernier transfert de fichier ([Envoie des csv](#)).