



## Livraison magasin

### Liste des fichiers

- AddOns.php
- images/
  - addons/
    - StoreDelivery/
      - mesrelais-point.png
- modeles/
  - css/
    - predefinedAddress.css
  - catalog/
    - cart/
      - StoreDelivery/
        - address.html
        - selector.html
        - styles.html
      - cart\_address.html
- specifs/
  - add-ons/
    - StoreDelivery/
      - actions/
        - GetFieldsEntry.php
        - Script.php
        - Search.php
      - class/
        - StoreDeliveryConfig.class.php
        - StoreDeliveryDirectory.class.php
        - StoreDeliveryHooks.class.php
        - StoreDeliveryModule.class.php
        - StoreDeliveryPoint.class.php
        - StoreDeliverySelector.class.php
        - directory/
          - Entry.php

- EntryDB.php
  - scripts/
    - layerconfig.js
    - list.js
  - strings/
    - strings\_fr.xml
  - predefined\_address\_tools/
    - scripts/
      - predefinedBehaviours.js
    - tools/
      - ConfigTableTools.class.php
      - LayerAbstract.class.php
      - MxToolsAbstract.php
    - PredefinedAddressConfig.class.php
    - PredefinedAddressConfigLayer.class.php
    - PredefinedAddressModule.class.php
    - PredefinedAddressPoint.class.php
    - PredefinedAddressSelector.class.php
- hooks/
  - catalog/
    - hook\_cart.class.php
    - hook\_shipping.class.php
  - core/
    - hook\_PublicSite.class.php
    - hook\_crud.class.php

## Fonctionnalité

- Livraison magasin
  - Intégration d'une liste de points de livraison magasin dans l'étape de sélection d'adresse du tunnel de commande. Les magasins sont définis par des fiches annuaires.
  - Possibilité d'activer ce service par règle de frais de port
  - Affichage des points relais sur une carte
  - Compatibilité avec d'autres modules de livraison d'adresse sur la même base.

## Pré-requis

- Version minimum d'E-majine : 1.17b
- Connaissances en intégration, javascript.
- Version de jquery minimum: 1.10

## Installation via FTP

1. Dézipper l'archive
2. S'il n'y est pas déjà présent, copier le fichier AddOns.php dans les répertoires distants /html/specifs/hooks/actions/public/ et /html/specifs/hooks/actions/manage
3. Copier le répertoire /specifs/add-ons dans le répertoire distant /specifs/libs/add-ons
4. Copier le répertoire images/addons/StoreDelivery/ dans le répertoire distant images/addons/StoreDelivery/
5. Copier le fichier /modeles/catalog/cart/StoreDelivery/ dans le répertoire distant /html/modeles/[langue]/[modèle]/catalog/cart/
6. Copier le répertoire /modeles/css/predefinedAddress.css dans le répertoire distant /html/modeles/[langue]/[modèle]/css/

## Gestion des hooks

L'add-on "StoreDelivery" utilise un système de [hooks](#) pour son bon fonctionnement. Le tableau suivant indique les hooks utilisés par l'add-on.

### - Si votre projet n'utilise pas de hooks

Si votre projet n'utilise pas de hooks, vous devez copier les répertoires /specifs/hooks/catalog et /specifs/hooks/core dans le dossiers specifs/hooks/ de votre projet.

### - Si votre projet utilise déjà ces hooks

Si votre projet contient déjà les hooks suivants, vous devez les modifier pour ajouter les éléments nécessaires à l'add-on comme indiqué dans le tableau suivant.

	Méthode	Code d'appel
<p>Fichier : <code>specif/hooks/catalog/hook_cart.class.php</code></p> <p><b>Insérer en haut de fichier les lignes suivantes :</b>  <code>require_once em_misc::getSpecifPath() . 'libs/add-ons/StoreDelivery/class/StoreDeliveryHooks.class.php';</code></p> <p><b>Déclaration de la classe :</b>  <code>class hook_cart extends Emajine_Hooks</code></p>		
1	public function onInit()	<code>\StoreDelivery\StoreDeliveryHooks::hook_cart_onInit();</code>
2	public function displayAddressStep(&\$mx)	<code>\StoreDelivery\StoreDeliveryHooks::hook_cart_displayAddressStep( \$mx );</code>
3	public function displayConfirmationAfterStandard(\$mx)	<code>\StoreDelivery\StoreDeliveryHooks::hook_cart_displayConfirmationAfterStandard( \$mx );</code>
4	public function saveOrderBeforeStandard(\$orderId, &\$orderDetail, &\$orderLines)	<code>\StoreDelivery\StoreDeliveryHooks::hook_cart_saveOrderBeforeStandard( \$orderId, \$orderDetail, \$orderLines );</code>
5	public function saveOrderAfterStandard(\$orderId, &\$orderDetail, &\$orderLines)	<code>\StoreDelivery\StoreDeliveryHooks::hook_cart_saveOrderAfterStandard(\$orderId, \$orderDetail, \$orderLines);</code>
<p>Fichier : <code>specif/hooks/catalog/hook_shipping.class.php</code></p> <p><b>Insérer en haut de fichier la ligne suivante :</b>  <code>require_once em_misc::getSpecifPath() . 'libs/add-ons/StoreDelivery/class/StoreDeliveryHooks.class.php';</code></p> <p><b>Déclaration de la classe :</b>  <code>class hook_shipping extends Emajine_Hooks</code></p>		
6	public function setAvailableShippingRules(&\$availableShippingRules)	<code>\StoreDelivery\StoreDeliveryHooks::hook_shipping_setAvailableShippingRules( \$availableShippingRules );</code>
<p>Fichier : <code>specif/hooks/core/hook_crud.class</code></p> <p><b>Insérer en haut de fichier la ligne suivante :</b>  <code>require_once em_misc::getSpecifPath() . 'libs/add-ons/StoreDelivery/class/StoreDeliveryHooks.class.php';</code></p> <p><b>Déclaration de la classe :</b>  <code>class hook_crud extends Emajine_Hooks</code></p>		
7	public function executeCrudActions(\$crudID, &\$crudObject, \$action)	<code>\StoreDelivery\StoreDeliveryHooks::hook_crud_executeCrudActions(\$crudID, \$crudObject, \$action);</code>
8	public function setLayerTabsAfterDefault(\$crudID, &\$crudObject, &\$layerObject)	<code>\StoreDelivery\StoreDeliveryHooks::hook_crud_setLayerTabsAfterDefault(\$crudID, \$crudObject, \$layerObject);</code>
<p>Fichier : <code>specif/hooks/core/hook_PublicSite.class</code></p>		

**Insérer en haut de fichier la ligne suivante :**

```
require_once em_misc::getSpecifPath() . 'libs/add-ons/StoreDelivery/class/StoreDeliveryHooks.class.php';
```

**Déclaration de la classe :**

```
class hook_PublicSite extends Emajine_Hooks
```

10	public function afterPageGeneration(\$content, \$context)	\StoreDelivery\StoreDeliveryHooks::hook_publicsite_afterPage Generation();
----	---	---

## Intégration Livraison magasin

**Attention** : Il est préférable, dans un premier temps d'effectuer l'intégration sur un modèle de test pour ne pas interférer sur le bon fonctionnement du tunnel de commande en production.

La seule partie d'intégration où vous allez devoir intervenir est dans le template de l'étape de sélection d'adresse dans le tunnel de commande : `cart_address.html`

Attention toutefois à quelques particularités qui doivent être respectées pour le bon fonctionnement de l'add-on.

- **Si votre site ne dispose pas encore de module de sélection d'adresses prédéfinies**

- Structure HTML de `cart_address.html` :

Sur le modèle du fichier de l'archive `/modeles/catalog/cart/cart_address.html`, il est impératif de reproduire la structure comprise dans le bloc mx "`<mx:bloc id="global">`".

Il est préférable de copier et d'intégrer tous les éléments compris entre la balise `<!-- SELECTION D'ADRESSE - BEGIN -->` et la balise `<!-- SELECTION D'ADRESSE - END -->`

Ces éléments comprennent :

- une redéfinition du processus de sélection d'adresse à domicile standard ( "`<mx:bloc id="delivery">`" )
- l'élément `addressSelector` qui affichera les éléments de la sélection d'adresses prédéfinies ("`<div id="addressSelector">`")
- un formulaire de recherche de points relais ("`<div class="search">`"). Il sera utilisé pour tous les modules de sélection d'adresse prédéfinies installés sur le site
- l'élément d'affichage de la carte (`<div class="map" id="predefinedAddressMap"></div>`)
- le conteneur affichant la liste de points relais (`<div id="availableModules">`). Il sera également utilisé pour tous les autres modules de sélection d'adresse prédéfinies installés sur le site

Attention à ce que les éléments suivants soient bien pris en compte pour un fonctionnement correct :

- l'élément form "adresseselectionvalidationform" doit englober le <mx:bloc id="delivery">
- un élément input doit être présent pour signifier le choix d'adresse de livraison à domicile tel que :

```
<input type="radio" mxAttribut="style:normalDeliveryStyle;" name="deliveryType" value="normalDelivery" id="normalDelivery" checked />
```

- le sélecteur de points relais doit être intégré dans le form "adresseselectionvalidationform" :

```
<!--[ADD_ON_StoreDelivery]-->
  <mx:bloc id="PredefinedAddress_StoreDelivery">
    <mx:text id="selector" />
  </mx:bloc id="PredefinedAddress_StoreDelivery">
<!--[/ADD_ON_StoreDelivery]-->
```

#### - Gestion du javascript :

Une fois la structure bien en place et conforme, vous allez pouvoir ajouter les éléments en javascript qui géreront les interactions. Ces éléments javascript seront eux aussi utilisés pour tous les autres modules de sélection d'adresse prédéfinies installés sur le site.

Dans le script type

(modeles/[langue]/[modèle]/catalog/cart/StoreDelivery/predefinedAddress.js), que vous pouvez modifier pour coller à vos besoins, vous retrouverez :

- La gestion de la suppression de l'affichage de la recherche et de la carte si aucun module de sélection d'adresses prédéfinies n'est activé.
- La gestion de la recherche ( launchSearch() )
- L'initialisation de l'écoute des événements d'interactions ( initStoreDelivery() )

Lorsque le module StoreDelivery est activé sur la page, un objet StoreDelivery est instancié. Cet objet permet d'écouter les éléments au lancement de la recherche ( StoreDelivery.onStartSearch( startSearch ) et StoreDelivery.onStopSearch( stopSearch ); ). Il permet aussi d'écouter des événements à la sélection de points relais, via la fonction :

```
StoreDelivery.onEvent(function(data){
  if( console && console.log ){
```

```

        console.log( data );
    }
})

```

Ci-dessous la liste des événements disponibles via la fonction onEvent :

Valeur de data.type	Autre champs de data	Déclenchement
selectAddress	data.id : id de l'adresse sélectionnée	à la sélection d'une adresse
unselectAddress	data.id : id de l'adresse sélectionnée	à la désélection d'une adresse
highlightAddress	data.id : id de l'adresse sélectionnée	au mouser de la popup ou du point de la liste
unlightAddress	data.id : id de l'adresse sélectionnée	Au mouseleave du point ou de la popup

**- Si votre site propose déjà d'autres modules de sélection d'adresses prédéfinies**

- Structure HTML de cart\_address.html :

La structure permettant la gestion de sélection d'adresses prédéfinies étant déjà en place, le nouvel add-on Livraison magasin ne nécessite pas d'intégration complexe. En effet, il vous suffit juste d'ajouter le code suivant dans le conteneur "<div id='availableModules'", à la suite ou avant les éléments existant du même type :

```

<!--[ADD_ON_StoreDelivery]-->
    <mx:bloc id="PredefinedAddress_StoreDelivery">
        <mx:text id="selector" />
    </mx:bloc id="PredefinedAddress_StoreDelivery">
<!--[/ADD_ON_StoreDelivery]-->

```

- Gestion du javascript :

De même pour la gestion du javascript, les éléments étant déjà en place, il suffit d'ajouter les appels à l'initialisation dans votre fichier de traitement js des add-ons de sélection d'adresses prédéfinies, par exemple :

```
//----- StoreDelivery -----  
if (typeof StoreDelivery != "undefined"){  
    initStoreDelivery();  
}  
  
function initStoreDelivery()  
{  
    StoreDelivery.onStartSearch( startSearch );  
    StoreDelivery.onStopSearch( stopSearch );  
    StoreDelivery.onEvent(function(data){  
        // if( console && console.log ){  
        //     console.log( data );  
        // }  
    });  
}
```

De même pour le lancement de la recherche, dans la fonction launchSearch(), par exemple :

```
if (typeof StoreDelivery != "undefined"){  
    StoreDelivery.search( "",  
        $("#addressSelector [name=research_cp]").val() ,  
        $("#addressSelector [name=research_ville]").val() );  
}
```

## Les templates modifiables :

Pour votre intégration, vous disposez de 2 templates modifiables permettant l’affichage du sélecteur de points relais et l’affichage d’un point relais dans le sélecteur et sur la carte. Les templates se trouvent dans /modeles/[langue]/[modèle]/catalog/cart/StoreDelivery/

- **selector.html** : ce template permet de gérer le sélecteur de points relais Livraison magasin.
- **address.html** : ce template permet de gérer l’affichage d’un point relais dans la liste, dans le bloc mx “<mx:bloc id="list">”. Il permet aussi l’affichage du point relais dans la carte via le bloc mx “<mx:bloc id="map">” (attention à respecter la structure de l’élément <mx:text id="mapPopup"/>).

Vous disposez aussi d’une feuille de style “type” dans ce répertoire, qui comprend une liste de règles css par défaut permettant un affichage correct du sélecteur.

Attention à ce que la règle suivante soit toujours disponible pour un fonctionnement optimal :

```
.StoreDelivery_hide{ display: none; }
```

## Utilisation de Livraison magasin

### Création d’une ou plusieurs type(s) de fiche annuaire

La liste est magasin proposé est constitué à partir des fiches annuaires.

Vous devez donc créer au moins un type de fiche annuaire avec obligatoirement une donnée de type géolocalisation. Pour cela rendez vous depuis le back-office d’E-majine dans l’écran de gestion des types de fiche (Annuaire > Configuration > Types de fiches) Chaque magasin aura pour nom le nom de la fiche annuaire et vous pourrez ensuite associer les données d’adresse via un mapping lors de la définition de la règle de frais de port

### Définition de la règle de frais de port

Pour associer une règle de frais de port au module de sélection Livraison magasin, vous devez vous rendre depuis le back-office d’E-majine dans l’écran de gestion des règles de frais de port ( Ma boutique > Configuration > Règles de frais de port ). Depuis cet écran, pour chaque règle, vous trouverez un nouvel onglet “Livraison magasin” d’où vous pourrez activer et configurer votre module.

Ci-dessous les actions disponibles :

- **Activation** : permet d’activer ou non le module de sélection d’adresses prédéfinies magasin pour la règle de frais de port courante. Si d’autres modules de sélection d’adresses prédéfinies sont installés sur le site, vous pouvez activer plusieurs modules pour une même règle de frais de port.
- **Configuration de l’étape d’adresses** : deux choix sont possibles :

- Proposer la livraison à domicile standard : L'utilisateur aura le choix entre la sélection de point relais et le processus standard de sélection d'adresse (domicile, travail, etc...)
- Ne pas proposer la livraison à domicile standard : Le sélecteur de points relais s'affichera seul dans l'étape d'adresse. L'utilisateur ne pourra que sélectionner un point relais.
- **Configuration de la liste des magasins** : Vous devez dans cette zone choisir le type de fiche annuaire qui constituera votre liste de magasin proposés par cette règle de frais de port. Vous devez également mapper les données du type de la fiche annuaire avec les données Adresse, Code postal, Ville et Pays. Ces données permettent de constituer l'adresse de livraison.

## Fonctionnement

Lorsqu'un utilisateur passe une commande, dans l'étape d'adresse, si le module Livraison magasin est activé pour la règle de frais de port courante (par défaut, la première règle de frais de port dans la liste de l'écran de configuration), l'utilisateur pourra choisir un magasin, soit en cliquant sur le marqueur de la carte, soit en cliquant sur un élément de la liste. Une fois le formulaire validé, il sera redirigé vers la page de confirmation de commande standard.

Sur la page de confirmation, une vérification sera faite sur la concordance entre la règle de frais de port et l'adresse de livraison. Autrement dit, l'utilisateur ne peut pas sélectionner un autre magasin pour une règle de frais de port n'ayant pas ce module activé. Et, à l'inverse, si la configuration "Ne pas proposer la livraison à domicile standard" est définie, l'utilisateur ne pourra pas sélectionner une adresse de livraison qui ne soit pas un magasin pour cette règle. En cas d'incohérence, l'utilisateur sera redirigé vers la page de sélection d'adresse.