

Module Magento SOON

Soon_AdvancedCache

Rédacteur

Hervé G.
Lead développeur Magento

herve@agence-soon.fr

AGENCE SOON
81 avenue du Bac
94210 LA VARENNE ST HILAIRE
Tel : +33 (0)1 48 83 95 96
Fax : +33 (0)1 41 81 35 67

Relecture & validation technique

Aurélien GOUTAL
Directeur technique

aurelien.goutal@agence-soon.fr

Historique des révisions

1.0	Version bêta 0.9.1 du module finalisée	05/12/2011
-----	--	------------

Sommaire

Sommaire	2
Introduction.....	3
Présentation du module.....	3
Objectif de ce document	3
Informations sur la licence	4
Crédits et copyrights	4
Installation.....	5
Prérequis	5
Installation du module	5
Vérification de l'installation du module.....	5
Configuration de base	6
Activer le cache de l'accueil	6
Gérer les délais avant expiration.....	7
Activer le cache de la page « Contacts » Activer le cache des tags populaires.....	8
Pages CMS à ne pas mettre en cache.....	8
Configuration avancée	9
Utiliser Symlink Cache	9
Utiliser Symlink Cache pour tout le cache (slow et fast backend)	9
Utiliser Symlink Cache uniquement pour le slow backend	10
Ajouter des blocs en cache.....	10
Exemple	12
De l'intérêt de la granularité apportée par Soon_AdvancedCache	14
Events Magento et caches de blocs	16
Exemple	16
Explication du code	18
Tableau des événements et objets	19

Introduction

Présentation du module

Soon_AdvancedCache offre différentes fonctionnalités dont l'objectif est double :

- accélérer sensiblement l'affichage des pages des boutiques Magento
- tout en allégeant les sollicitations au serveur qui effectue ainsi moins de calculs PHP et de requêtes MySQL.

Eléments mis en cache par Soon_AdvancedCache :

- Page d'accueil (à activer – voir rubriques suivantes)
- Pages CMS
- Blocs CMS
- Fil d'ariane
- Page catégorie et listings produits (y compris résultats de recherche)
- Navigation par filtre
- Fiche produit
- Tags populaires (à activer – voir rubriques suivantes)
- ... et tout autre bloc du layout moyennant quelques configurations voire quelques légers développements.

Objectif de ce document

Le présent document a pour objectif de vous présenter le module SOON que vous venez de télécharger et vous guider dans le processus d'installation, paramétrage et utilisation de ce dernier.

Légendes du document :



Complément d'information



Complément d'information important



Complément d'information disponible sur Internet en suivant le lien affiché à droite de cette icône.

Informations sur la licence

Cette extension est mise à disposition de la communauté sous la licence Open Source « APL 1.0 » (Adaptive Public Licence) donc le détail est consultable à l'adresse suivante :

<http://www.opensource.org/licenses/APL-1.0>

Toute utilisation de cette extension est soumise à l'acceptation des termes de la licence AFL 1.0.

Crédits et copyrights

Cette extension pour Magento a été développée par l'Agence-Soon (NOVATIX SARL) afin d'être mise à disposition gratuitement pour la communauté Magento. L'Agence-Soon peut être contacté pour toute question en rapport avec cette extension à l'adresse magento@agence-soon.fr.

L'extension, son code, sa documentation et tout autre matériel venant avec la présente extension que Soon aurait réalisé reste la propriété exclusive de l'Agence-Soon qui en détient tous les droits.

Installation

Prérequis

- Magento avec un core non modifié ;
- Magento en version 1.4.x, 1.5.x ou 1.6.x ;
- **Module Soon All installé – voir http://www.agence-soon.fr/soon_all**



D'autres modules agissant sur le cache peuvent influencer ou compromettre le bon fonctionnement de Soon_AdvancedCache et réciproquement.



Bien que l'effet de Soon_AdvancedCache soit visible sur votre boutique Magento sans configuration spécifique, des compétences en développement Magento restent obligatoires afin d'utiliser cet outil à son meilleur potentiel et d'en adapter le fonctionnement à votre projet.

Installation du module

L'installation est une installation standard soit par Magento Connect soit par dépôt sur votre serveur des fichiers contenus dans l'archive téléchargée sur notre site.



Veillez à copier le fichier de layout
/app/design/frontend/default/default/layout/advancedcache.xml
dans votre dossier de thème.

Vérification de l'installation du module

Afin de vérifier l'installation correcte du module veuillez :

1. Purger totalement le cache Magento :
 - a. « Système > Configuration > Gestion du cache »
 - b. Cliquez sur le bouton « Purger le cache Magento »
2. Déconnectez-vous du backoffice puis reconnectez-vous ;
3. Rendez-vous dans « Système > Configuration »
4. Cliquez sur « SOON > Gestion avancée du cache » dans le menu de gauche.

Si la page de présentation du module Soon_AdvancedCache s'affiche, le module a bien été installé.

Configuration de base

Activer le cache de l'accueil

Mettre en cache le contenu de la page d'accueil permet de s'affranchir de sa génération dynamique à chaque consultation.

Pour activer le cache de la page d'accueil :

Rendez-vous dans « Système > Gestion du cache »

Puis activer le cache « Homepage | Blocks de la page d'accueil | HOMEPAGE_BLOCKS »

Gestion du cache Purger le cache Magento Purger le stockage du cache

Sélectionner tout |
 Désélectionner tout |
 Sélectionner les éléments visibles |
 Désélectionner les éléments visibles |
 0 élément(s) sélectionné(s)

Actions: Rafraîchir Valider

Type de Cache	Description	Tags associés	Statut
<input type="checkbox"/> Configuration	Système (config.xml, local.xml) et configuration de modules (config.xml).	CONFIG	ACTIVE
<input type="checkbox"/> Agencements	Instructions de construction d'agencement.	LAYOUT_GENERAL_CACHE_TAG	ACTIVE
<input type="checkbox"/> Sortie de blocs HTML	Code HTML des blocs de page.	BLOCK_HTML	ACTIVE
<input type="checkbox"/> Traductions	Fichiers de traduction.	TRANSLATE	ACTIVE
<input type="checkbox"/> Données de collections	Fichiers de données de collection.	COLLECTION_DATA	ACTIVE
<input type="checkbox"/> Types EAV et attributs	Cache de déclaration des types d'entité.	EAV	ACTIVE
<input type="checkbox"/> Configuration des Web Services	Fichiers de définition des Web Services (api.xml).	CONFIG_API	ACTIVE
<input type="checkbox"/> Homepage	Blocks de la page d'accueil	HOMEPAGE_BLOCKS	DÉSACTIVÉ

Gestion du cache Purger le cache Magento Purger le stockage du cache

Sélectionner tout |
 Désélectionner tout |
 Sélectionner les éléments visibles |
 Désélectionner les éléments visibles |
 0 élément(s) sélectionné(s)

Actions: Rafraîchir Valider

Type de Cache	Description	Tags associés	Statut
<input type="checkbox"/> Configuration	Système (config.xml, local.xml) et configuration de modules (config.xml).	CONFIG	ACTIVE
<input type="checkbox"/> Agencements	Instructions de construction d'agencement.	LAYOUT_GENERAL_CACHE_TAG	ACTIVE
<input type="checkbox"/> Sortie de blocs HTML	Code HTML des blocs de page.	BLOCK_HTML	ACTIVE
<input type="checkbox"/> Traductions	Fichiers de traduction.	TRANSLATE	ACTIVE
<input type="checkbox"/> Données de collections	Fichiers de données de collection.	COLLECTION_DATA	ACTIVE
<input type="checkbox"/> Types EAV et attributs	Cache de déclaration des types d'entité.	EAV	ACTIVE
<input type="checkbox"/> Configuration des Web Services	Fichiers de définition des Web Services (api.xml).	CONFIG_API	ACTIVE
<input type="checkbox"/> Homepage	Blocks de la page d'accueil	HOMEPAGE_BLOCKS	ACTIVE

Statuts du cache de la page d'accueil.

Gérer les délais avant expiration

Soon_AdvancedCache offre la possibilité de choisir des délais avant expiration des caches en fonction de leur type : contenu ou catalogue. Par défaut, ces deux délais sont fixés à 7200 secondes (2 heures) mais vous choisirez peut-être d'affecter un délai plus long pour le contenu (pages et blocs CMS) car sa fréquence de mise à jour est souvent plus faible que celle des pages du catalogue.

Pour modifier les délais avant expiration :

Rendez-vous dans « Système > Gestion du cache »

Puis dans la zone « Gestion des délais d'expiration »

Saisir les nouvelles valeurs puis sauvegarder grâce au bouton prévu à cet effet



Gestion des délais d'expiration		Sauvegarder
Pages et blocs CMS	<input type="text" value="7200"/> secondes	
Catalogue (produits et catégories)	<input type="text" value="7200"/> secondes	

Délais d'expiration.

Activer le cache des tags populaires

A des fins pédagogiques pour la suite de cette présentation et la prise en main plus avancée de Soon_AdvancedCache, la mise en cache des Tags Populaires n'est pas faite automatiquement. Voici donc comment l'activer.

Pour activer le cache « Tags populaires » :

Rendez-vous dans « Système > Gestion avancée du cache > Cache de blocs »

Dans la grille, sélectionner la ligne déjà présente (ID 1)

Puis, à l'instar de la gestion du cache native de Magento, choisir « Activer » dans la liste déroulante prévue à cet effet sans oublier de valider.

La diode du cache « Tags Populaires » doit maintenant être passée de « Désactivé » à « Activé »

Pages à ne pas mettre en cache (exceptions)

Selon votre projet, il peut être nécessaire de créer des exception afin de ne pas mettre en cache le contenu de certaines pages CMS. Par défaut, la page 404 n'est pas mise en cache et est donc présente dans la liste des exceptions lors de l'installation de Soon_AdvancedCache.

Pour ajouter une exception :

Rendez-vous dans « Système > Gestion avancée du cache > Exceptions »

Cliquer sur « Ajouter »

Choisir la page CMS à ne pas mettre en cache grâce à la liste déroulante proposée

Enregistrer l'exception



Pour modifier ou supprimer une exception :

Utiliser la liste déroulante présente pour chaque ligne de la grille des exceptions.



Configuration avancée

Utiliser Symlink Cache

D'après les comparatifs réalisés par Vinai Kopp sur le blog Magebase, l'utilisation de liens symboliques pour le dossier var/cache de l'instance Magento apporte de réels gains de performances. Nous avons donc intégré cette fonctionnalité à Soon_AdvancedCache.



Symlink Cache ne fonctionne que sur des environnements Unix. Le Symlink Cache a été mis en place avec succès par l'Agence Soon sur plusieurs projets utilisant une distribution basée Debian.



Etudes et analyses de Vinai Kopp sur le blog Magebase : <http://magebase.com/magento-tutorials/improving-the-file-cache-backend/>

Utiliser Symlink Cache pour tout le cache (slow et fast backend)

- Créer une sauvegarde de secours de app/etc/local.xml
- Editer app/etc/local.xml
- Chercher

```
<session_save><![CDATA[files]]></session_save>
```

- Remplacer par

```
<session_save><![CDATA[files]]></session_save>

<cache>
  <backend>Soon_AdvancedCache_Model_Symlink</backend>
  <backend_options>
    <cache_dir>var/cache</cache_dir>
    <hashed_directory_level>1</hashed_directory_level>
    <hashed_directory_umask>0777</hashed_directory_umask>
    <file_name_prefix>mage</file_name_prefix>
  </backend_options>
</cache>
```

- Rendez-vous dans « Système > Gestion avancée du cache > Symlink Cache »
- Initialiser les symlinks en cliquant sur « Initialiser les symlinks »

Symlink Cache Initialiser les symlinks

Statut

Le Symlink Cache est actuellement **actif**
Fast Cache Backend: Soon_AdvancedCache_Model_Symlink

Symlink Cache correctement configuré dans local.xml : le bouton « Initialiser les Symlinks » apparaît.

Utiliser Symlink Cache uniquement pour le slow backend

Si vous désirez utiliser un outil tiers type APC ou Memcached pour gérer le fast backend mais préférez Symlink Cache pour le slow backend :

- S'assurer que le système de cache mis en place accepte l'utilisation d'un slow backend
- Créer une sauvegarde de secours de app/etc/local.xml
- Editer app/etc/local.xml
- Modifier / insérer la ligne suivante dans le noeud <config><global><cache>

```
<slow_backend>Soon_AdvancedCache_Model_Symlink</slow_backend>
```

- Rendez-vous dans « Système > Gestion avancée du cache > Symlink Cache »
- Initialiser les symlinks en cliquant sur « Initialiser les symlinks »

Ajouter des blocs en cache

Soon_AdvancedCache est capable de mettre en cache n'importe quel bloc issu du layout XML (block).

Pour ajouter un bloc en cache :

Rendez-vous dans « Système > Gestion avancée du cache > Caches de blocs »

Créer un nouveau cache de bloc en cliquant sur le bouton « Ajouter »

Renseigner les champs comme suit :

- **Identifiant (Tag) :** le tag du cache. Doit être unique car chaque bloc est mis en cache séparément. On préférera une nomenclature type « minuscules avec underscores ».
- **Classe de bloc :** la classe « Block » générant le bloc dans la vue. Doit être unique car un bloc (donc sa classe associée) ne peut être mis en cache qu'une fois.
- **Nom du bloc dans le layout :** correspond à l'attribut « name » du bloc dans le layout XML.
- **Description :** complément d'information au sujet du bloc mis en cache.
- **Expiration :** choisir un des délais proposés. Les délais « Pages et bloc CMS » et « Catalogue » sont ceux configurés dans « Système > Gestion du cache » puis dans la zone « Gestion des délais d'expiration ».
- **Configuration spéciale :** selon le code et l'usage du « Block » que vous souhaitez mettre en cache, il peut être nécessaire de compléter la configuration de sa mise en cache.
 - Dépend de la catégorie courante
Si l'information contenue dans le bloc est fonction de la catégorie en cours de consultation. Le bloc que vous tenterez ici de mettre en cache aura donc de fortes chances d'avoir une ou plusieurs des caractéristiques suivantes :
 - Appel sur les pages catégories dans le layout XML :
catalog_category_default et/ou catalog_category_layered par exemple
 - Récupération de la catégorie courante :
Mage::registry('current_category') et/ou \$this->getCurrentCategory() par exemple

- Dépend du produit courant
Scénario identique à « Dépend de la catégorie courante » mais avec un produit.
 - Appel sur la fiche produit dans le layout XML :
catalog_product_view
 - Récupération du produit courant :
Mage ::registry('current_product') et/ou \$this->getProduct()
- Est une réécriture de Products List / Product View / Breadcrumbs
Il est ici nécessaire de préciser si le bloc que vous tentez de mettre en cache est une réécriture (rewrite) d'une des classes ci-dessous :
 - Mage_Catalog_Block_Product_List
/app/code/core/Mage/Catalog/Block/Product/List.php
 - Mage_Catalog_Block_Product_View
/app/code/core/Mage/Catalog/Block/Product/View.php
 - Mage_Page_Block_Html_Breadcrumbs
/app/code/core/Mage/Page/Block/Html/Breadcrumbs.php
- **Statut** : faut-il activer la mise en cache du bloc ?



Il est possible d'ajouter des blocs en cache par le code. Cette documentation ne traite pas de cette méthode. Veuillez-vous référer à la classe `Soon_AdvancedCache_Model_Project_Add`.

Exemple

Objectif : mettre en cache un bloc présent en colonnage de gauche et contenant la liste des sous-catégories de la catégorie en cours de consultation.



Le nouveau bloc « Children Categories ».



Dans cet exemple, le module créé et utilisé pour accueillir le bloc à mettre en cache est « Soon_MageOverload ».

1. Création d'une classe de block dans le module local (Soon_MageOverload dans cet exemple) On notera l'extension de la classe Mage_Catalog_Block_Navigation nécessaire afin que Magento ne serve pas la version mise en cache du menu de navigation mais bel et bien un nouveau bloc.

```
class Soon_MageOverload_Block_Children extends Mage_Catalog_Block_Navigation {
    public function getCurrentChildCategories() {
        return parent::getCurrentChildCategories();
    }
}
```

2. Appel à ce bloc dans le fichier de layout local.xml du thème

```
<catalog_category_default>
  <reference name="left">
    <block type="mageoverload/children" name="children-categories" before="-"
template="catalog/navigation/children-categories.phtml"/>
  </reference>
</catalog_category_default>

<catalog_category_layered>
  <reference name="left">
    <block type="mageoverload/children" name="children-categories" before="-"
template="catalog/navigation/children-categories.phtml"/>
  </reference>
</catalog_category_layered>
```

3. Fichier de template catalog/navigation/children-categories.phtml

```
<?php $childrenCategories = $this->getCurrentChildCategories(); ?>
<?php if($childrenCategories->count() > 0) : ?>
  <div class="block block-poll">
    <!-- class .block-poll pour profiter d'une css existante -->
    <div class="block-title">
      <strong><span><?php echo $this->__('Children Categories'); ?></span></strong>
    </div>
    <div class="block-content">
      <ul>
        <?php foreach($childrenCategories as $child) : ?>
          <li><?php echo $child->getName() ?></li>
        <?php endforeach; ?>
      </ul>
    </div>
  </div>
</div>
<?php endif; ?>
```

4. Configuration du cache de bloc Soon_AdvancedCache (voir page suivante)
5. Rafraîchir les caches de layout – « Système > Gestion du cache > Agencements »
6. Rafraîchir une page catégorie du front-office pour afficher le nouveau bloc « Children Categories »

Éditer le bloc 'children_categories'

[Retour](#) [Réinitialiser](#) [Supprimer le bloc](#) [Sauvegarder le bloc](#)

Block information

Identifiant (Tag) *	<input type="text" value="children_categories"/>
Classe de bloc *	<input type="text" value="Soon_MageOverload_Block_Children"/>
Nom du bloc dans le layout *	<input type="text" value="children-categories"/>
Description	<input type="text" value="Liste des sous-catégories de la catégorie en cours de consultation"/>
Expiration	<input type="text" value="Catalogue (produits et catégories)"/>
Configuration spéciale	<input type="text" value="Dépend de la catégorie courante"/>
Statut	<input type="text" value="Activé"/>

Configuration du cache bloc « children_categories »

Intérêt de la granularité apportée par Soon_AdvancedCache

A partir de l'exemple ci-dessus, nous désirons modifier le template afin d'afficher le nombre d'articles présents dans chacune des sous-catégories.

Dans le fichier de template catalog/navigation/children-categories.phtml, la ligne

```
<li><?php echo $child->getName(); ?></li>
```

devient

```
<li><?php echo $child->getName() ?> (<?php echo $child->getProductCount() ?>)</li>
```

Sur une boutique Magento non équipée de Soon_AdvancedCache, il est obligatoire de rafraîchir l'ensemble des sorties de blocs HTML, y compris ceux qui n'ont pas besoin de l'être. Ceci a pour conséquence d'impacter les ressources du serveur qui va devoir régénérer le cache de blocs dont le contenu n'a pas changé et, par conséquent, le délai d'affichage des pages s'en trouvera allongé.

Avec Soon_AdvancedCache, il est possible de ne rafraîchir que les blocs devant l'être et ainsi d'économiser des ressources serveur.

Dans notre exemple, suite à l'ajout du nombre d'articles présents dans chaque sous-catégorie, nous allons rafraîchir uniquement le bloc de cache associé (identifiant children_categories).

Pour rafraîchir un ou plusieurs blocs :

Rendez-vous dans « Système > Gestion avancée du cache > Caches de blocs »

Editer le bloc « children_categories » de notre exemple

Vider le cache en cliquant sur le bouton « Vider le cache »

Rafraîchir le site en front-office et constater la mise à jour du bloc



Il est possible de rafraîchir plusieurs blocs en masse. Le fonctionnement est identique au cache natif de Magento :

Rendez-vous dans « Système > Gestion avancée du cache > Caches de blocs »

Sélectionner les blocs à rafraîchir

Choisir « Rafraîchir » dans la liste déroulante en en-tête de grille puis valider.



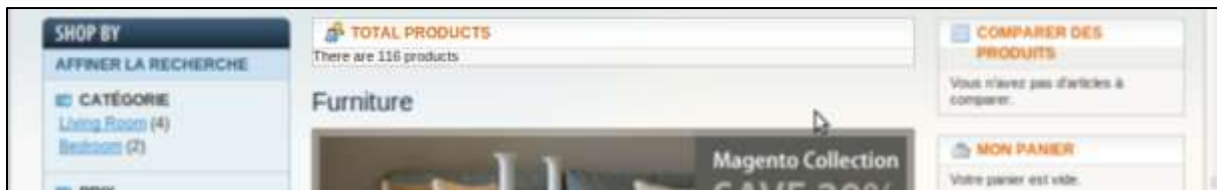
Le bloc « Children Categories » mis à jour avec le nombre d'articles entre parenthèses.

Events Magento et caches de blocs

Voici comment vider vos caches de blocs en fonction de certains événements.

Exemple

Objectif : mettre en cache et rafraîchir automatiquement un bloc présent en colonne principale (content) et contenant le nombre totaux de produits.



Le nouveau bloc « Total Products ».



Dans cet exemple, le module créé et utilisé pour accueillir le bloc à mettre en cache est « Soon_MageOverload ».

1. Création d'une classe de block dans le module local (Soon_MageOverload dans cet exemple).

```
class Soon_MageOverload_Block_Count extends Mage_Core_Block_Template {  
  
    public function getProductsCount() {  
        return (int) Mage::getResourceModel('catalog/product_collection')->count();  
    }  
}
```

2. Appel à ce bloc dans le fichier de layout local.xml du thème

```
<default>  
    <reference name="content">  
        <block type="mageoverload/count" name="products-count" before="-"  
            template="catalog/products-count.phtml"/>  
    </reference>  
</default>
```

3. Fichier de template catalog/products-count.phtml

```
<div class="block block-poll">
<!-- class .block-poll pour profiter d'une css existante -->
  <div class="block-title">
    <strong><span><?php echo $this->__('Total products'); ?></span></strong>
  </div>
  <div class="block-content">
    <?php echo $this->__('There are %s products', $this->getProductsCount()); ?>
  </div>
</div>
```

4. Configuration du cache de bloc Soon_AdvancedCache

The screenshot shows the 'Éditer le bloc 'products_count'' configuration page. At the top, there are buttons for 'Retour', 'Réinitialiser', 'Supprimer le bloc', and 'Sauvegarder le b'. Below this is the 'Block information' section with the following fields:

Identifiant (Tag) *	products_count
Classe de bloc *	Soon_MageOverload_Block_Count
Nom du bloc dans le layout *	products-count
Description	Nombre total de produits
Expiration	Catalogue (produits et catégories) ▼
Configuration spéciale	▼
Statut	Activé ▼

5. Rafraîchir le front-office pour afficher les modifications et le nouveau bloc « Total Products »

Le bloc est maintenant créé et mis en cache mais constatons son comportement en cas de suppression de produit du catalogue par exemple.

6. Rendez-vous dans le back-office et supprimer un article
7. Rafraîchir le front-office sans toucher aux caches
8. Constaté que le nombre de produits n'a pas évolué dans le bloc « Total Products »

Il faut donc rafraîchir le cache du bloc dont l'identifiant (Tag) créé plus haut est « products_count » dès qu'un article est supprimé (ou ajouté) au catalogue.

Pour rafraîchir un cache de bloc lors d'un event :

1. Créer une copie de sauvegarde et éditer le fichier
/app/code/community/Soon/AdvancedCache/Model/Project/Clean.php
2. Chercher la ligne

```
// Additional code
```

3. Remplacer par :

```
// Additional code
if (($object instanceof Mage_Catalog_Model_Product) === true) {
    $projectTagsToClean[] = 'products_count';
}
```

4. Rafraîchir le cache du bloc dont l'identifiant (Tag) est « products_count »
5. Rafraîchir le front-office et constater la mise à jour du bloc « Total Products »
6. Supprimer et ajouter des produits à loisir au catalogue. Le bloc « Total Products » restera à jour.

Explication du code

```
// Additional code
if (($object instanceof Mage_Catalog_Model_Product) === true) {
    $projectTagsToClean[] = 'products_count';
}
```

- Fichier : /app/code/community/Soon/AdvancedCache/Model/Project/Clean.php
- Classe : Soon_AdvancedCache_Model_Project_Clean
- Méthode : _populateTagsToClean(\$object, \$observer = null)

La méthode « _populateTagsToClean » est appelée sur plusieurs events Magento (voir plus bas *Tableau des événements et objets*).

Elle permet d'ajouter des tags de caches devant être vidés lors de l'intervention de ces events.

Ces tags sont regroupés dans la table \$projectTagsToClean

Le code permet de peupler cette table.

Dans l'exemple : si l'objet ayant généré l'événement est une instance de « Mage_Catalog_Model_Product », nous ajoutons le cache de bloc dont l'identifiant (Tag) est « products_count » à la liste des caches à vider (\$projectTagsToClean).



\$projectTagsToClean accepte :

- les tags utilisés par Magento
- les tags associés à des caches blocs créés dans l'admin (voir « Ajouter des blocs en cache », page 10)
- les tags des blocs mis en cache automatiquement par Soon_AdvancedCache (voir la classe Soon_AdvancedCache_Model_Add pour comprendre leur construction)

Tableau des événements et objets

Libellé	Event	Objet
Enregistrement catégorie en BO	catalog_category_save_before	Mage_Catalog_Model_Category
Suppression catégorie en BO	catalog_category_delete_before	Mage_Catalog_Model_Category
Enregistrement produit en FO	catalog_product_save_before	Mage_Catalog_Model_Product
Enregistrement produit en BO	catalog_product_save_before	Mage_Catalog_Model_Product
Suppression produit en BO	catalog_product_delete_before	Mage_Catalog_Model_Product
Enregistrement page CMS en BO	cms_page_save_before	Mage_Cms_Model_Page
Suppression page CMS en BO	cms_page_delete_before	Mage_Cms_Model_Page

Libre à vous d'adapter vos modules ou le fichier config.xml de Soon_AdvancedCache. Vous pouvez ainsi observer d'autres événements et modifier Soon_AdvancedCache_Model_Project_Clean pour déclencher le rafraîchissement de caches blocs en fonction de ces nouveaux événements.