

# OVIDENTIA

## Gestion des skins sous Ovidentia





<b>Version Documentation</b>	<b>Version OVIDENTIA / Module</b>	<b>Date</b>
1.0	6.5.2 / -	06/02/2008
1.1	6.5.2 / -	11/02/2008
1.2	7.1.7 / -	01/12/2009

## Table des matières

1 - Introduction.....	4
2 - Présentation.....	4
2.1 - Description.....	4
2.2 - Structure d'un skin.....	4
3 - Administration des skins dans OVIDENTIA.....	5
3.1 - Administration.....	5
3.2 - Options de l'utilisateur.....	5
4 - Créer son skin.....	6
4.1 - Conseils.....	6
4.2 - Fonctionnement dans le noyau.....	6
4.3 - Le moteur de templates.....	6
4.4 - Le langage OVML.....	7
4.5 - Méthode de personnalisation d'un skin.....	7
5 - Le moteur de templates en détails.....	8
5.1 - Syntaxe.....	8
5.2 - Les patrons d'affichage.....	9
6 - Le langage OVML en détails.....	10
6.1 - Syntaxe.....	10
6.2 - Pages d'accueil.....	11
7 - Annexes.....	12
7.1 - Les variables globales.....	12
7.2 - Les fichiers templates modifiables.....	13
7.3 - Les variables utilisables dans mailinfo.html.....	13
7.3.1 - Pour personnaliser l'objet de l'email des notifications de nouveaux articles.....	13
7.3.2 - Pour personnaliser l'objet de l'email des notifications de mise à jour des articles.....	14
7.3.3 - Pour personnaliser l'objet de l'email des notifications des approbateurs d'un article.....	14
7.3.4 - Pour personnaliser l'objet de l'email des notifications pour les commentaires.....	14
7.3.5 - Pour personnaliser l'objet de l'email des notifications de d'acceptation/refus d'un article.....	15
7.3.6 - Pour personnaliser l'objet de l'email des notifications des nouvelles contributions de forum.....	15

# 1 - Introduction

Cette documentation a pour but d'expliquer le fonctionnement des skins dans **OVIDENTIA** et leurs personnalisations par le webmaster.

## 2 - Présentation

### 2.1 - Description



Un skin est le terme utilisé dans **OVIDENTIA** pour désigner un thème graphique. On dit d'un skin qu'il est l'habillage d'un site (skin = peau en anglais), c'est à dire une couche graphique qui n'influe pas sur le traitement des données. Il rassemble tous les éléments graphiques et les règles d'affichage : images, mises en forme (fichiers CSS), templates, modèles d'affichage...

Dans **OVIDENTIA**, un skin correspond à un répertoire présent sur le serveur Web dans le répertoire skins à la racine du site :



racine d'**OVIDENTIA**

**OVIDENTIA** propose généralement 2 skins dans son installation : un skin nommé 'ovidentia' et un skin nommé 'ovidentia\_sw'. Il est conseillé de prendre pour base le skin 'ovidentia\_sw' pour une personnalisation.

### 2.2 - Structure d'un skin

Afin d'être reconnu comme un skin dans l'administration du site et afin de personnaliser les templates, les pages d'accueil et les styles, un skin doit contenir les répertoires suivants :

- images : par convention, ce répertoire rassemble toutes les images (bandeau, puces...)
- ovml : rassemble tous les fichiers du langage OVML (private.html, public.html...)
- styles : contient les fichiers css styles (ovidentia.css)
- templates : contient les fichiers templates personnalisés (page.html, sectiontemplate.html...)

Remarque : seuls les répertoires ovml, styles et templates sont nécessaires au fonctionnement d'un skin, d'autres répertoires peuvent être ajoutés pour séparer les applications flash des librairies javascript, etc...

## 3 - Administration des skins dans OVIDENTIA

### 3.1 - Administration

L'administrateur définit dans la configuration du site (Section Administration/Sites/nom du site/Configuration du site) le skin qui sera utilisé par les utilisateurs anonymes.

Nom de site	<input type="text" value="Ovidentia"/>
Description	<input type="text" value="Ovidentia site"/>
Slogan du site	<input type="text" value="Open Source CMS"/>
Langue	<input type="text" value="fr"/>
Filtre de la langue	<input type="text" value="Pas de filtre"/>
Skin	<input type="text" value="ovidentia_sw"/> <input type="text" value="ovidentia.css"/>
Adresse de messagerie du site	<input type="text" value="admin@your-domain.com"/>
Nom à utiliser pour les emails de notifications	<input type="text" value="Ovidentia Administrator"/>
Activer les statistiques?	<input type="text" value="Non"/>
Composition du nom d'utilisateur	<input type="text" value="Nom Prénom"/>
<input type="button" value="Enregistrer"/>	

Configuration du site

Les skins présents sur le serveur apparaissent dans la liste déroulante ainsi qu'une liste des styles css associés à chaque skin. En effet, un skin peut proposer plusieurs feuilles css afin de gérer des modifications minimales de couleurs, des tailles de polices différentes...

Remarque :

Cette configuration n'a d'effet que sur les pages en mode anonyme. En mode authentifié, chaque utilisateur a la possibilité de choisir son skin depuis ces options personnelles (voir ci-dessous). Pour supprimer cette option et ainsi empêcher le choix du skin par les utilisateurs, l'administrateur peut désactiver l'option 'L'utilisateur peut modifier son skin' dans la configuration du site (Section Administration/sites/nom du site/Options de l'utilisateur).

### 3.2 - Options de l'utilisateur

En section Utilisateur, lien Options, l'utilisateur a la possibilité de définir le skin qui sera utilisé pendant sa navigation sur le site en mode authentifié. Cette possibilité est rendue active par l'administrateur.

## 4 - Créer son skin

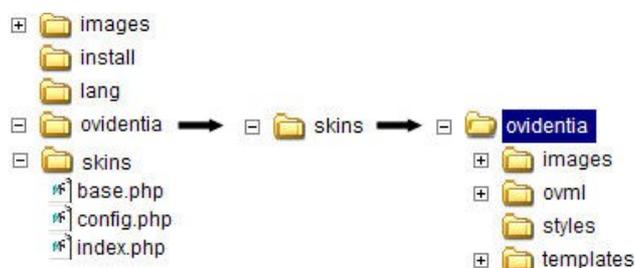
Pour créer son propre skin, il est conseillé de prendre comme base le skin fourni dans l'installation d'**OVIDENTIA** 'ovidentia\_sw'. Partant de cette base, il suffit dans un premier temps de le renommer et de le placer au même niveau qu'ovidentia\_sw sur le serveur Web pour qu'il soit reconnu dans l'administration d'**OVIDENTIA**.

### 4.1 - Conseils

- **OVIDENTIA** utilise la norme XHTML et le doctype 'XHTML 1.0 Transitional' pour toutes ces pages Web. Veuillez à respecter cette norme ; des outils peuvent valider vos pages (ex. : <http://validator.w3.org>).
- Veuillez à ne mettre aucune référence en absolue dans vos pages Web : le nom de votre skin ou l'adresse Web de votre site ne doit pas figurer dans votre code. Dans le cas contraire, la modification future du nom de domaine de votre site ou du nom du skin causerait des problèmes.  
Exemple avec une image :  
Syntaxe correcte : ``  
Syntaxe incorrecte : ``  
Ici, la variable globale babSkin est utilisée pour remonter le nom du skin courant dans un fichier template.
- Afin de rester compatible avec les dernières évolutions d'**OVIDENTIA**, les fichiers templates dits 'non modifiables' ne doivent pas être modifiés. Voir la liste des fichiers templates en annexe pour plus de détails.

### 4.2 - Fonctionnement dans le noyau

Le noyau d'**OVIDENTIA** contient un skin de base. Ce skin se nomme 'ovidentia' et rassemble la feuille de styles principale et les fichiers templates gérant toutes les interfaces.



skin de base dans le noyau

Pour que les skins personnalisés ne soient pas supprimés après chaque mise à jour du noyau **OVIDENTIA**, ces skins sont placés à la racine du site.

Remarque : faites attention à ne pas confondre les répertoires 'skins' à la racine du site et dans le noyau.

### 4.3 - Le moteur de templates

**OVIDENTIA** utilise un moteur de templates pour gérer toutes ses interfaces. Son principal avantage est de séparer le traitement des données (codes PHP, requêtes SQL...) de leurs visualisations (balises html, styles css...). Ainsi un écran d'**OVIDENTIA** correspond généralement à un fichier template. Un mini-langage de programmation permet de faire la liaison entre les fichiers PHP et les fichiers HTML.

## 4.4 - Le langage OVML

Le langage OVML est propre à **OVIDENTIA**. Il est rarement utilisé dans le noyau car il est destiné au webmaster afin de lui permettre de dynamiser ses pages. Le noyau fournit une librairie complète sur le contenu d'**OVIDENTIA** (articles, faqs, forums, fichiers...). Ainsi les pages d'accueil peuvent être dynamiques en affichant par OVML la liste des 10 derniers articles publiés.

## 4.5 - Méthode de personnalisation d'un skin

Le webmaster a la possibilité de modifier les styles css et les fichiers templates (interfaces). Sans personnalisations, **OVIDENTIA** utilise le skin de base contenu dans son noyau.

Personnaliser une interface d'**OVIDENTIA** revient donc à modifier le fichier template concerné. Afin qu'aucune modification ne soit perdue après mise à jour du noyau, le fichier template du noyau doit être copié dans le skin présent à la racine du site. Avant chaque affichage, **OVIDENTIA** vérifie donc si le fichier template concerné existe dans le skin choisi par l'utilisateur dans quel cas ce fichier sera utilisé comme affichage. Dans le cas contraire, **OVIDENTIA** utilisera le fichier template présent dans le noyau.

- Modifier la feuille de styles :

Il est conseillé de prendre comme base la feuille de styles 'ovidentia.css' du skin 'ovidentia\_sw' car elle contient les styles les plus couramment utilisés.

La feuille de styles CSS doit être placée dans le répertoire 'styles' de votre skin.

Remarque :

- Vous pouvez imbriquer des feuilles de styles. Cependant, évitez de placer tous les fichiers css dans le répertoire 'styles' de votre skin car dans les interfaces d'administration, **OVIDENTIA** liste et propose tous les fichiers css de ce répertoire.
- Évitez d'appliquer des styles trop généraux sur les éléments HTML afin de ne pas perturber l'affichage des interfaces.

- Modifier un fichier template :

Pour des raisons logiques d'évolution du portail, il n'est pas possible de modifier toutes les interfaces d'**OVIDENTIA** (fichiers templates). Ainsi, seuls quelques fichiers templates sont modifiables par le webmaster (voir la liste en annexe). La structure de ces fichiers ne change pas au fur et à mesure des versions d'**OVIDENTIA** ce qui garantit au webmaster que son skin restera compatible avec les futures versions d'**OVIDENTIA**.

Pour exemple, le fichier template sectiontemplate.html gère l'affichage des sections. Si vous désirez modifier l'apparence des sections, vous pouvez simplement modifier la feuille de styles du skin. Si vous avez besoin de modifier le balisage HTML des sections, procédez aux opérations suivantes :

- copier le fichier sectiontemplate.html du noyau (racine d'Ovidentia/noyau/skins/ovidentia/templates/) dans votre skin (racine d'Ovidentia/skins/votreskin/templates)
- modifiez le fichier sectiontemplate.html de votre skin

## 5 - Le moteur de templates en détails

### 5.1 - Syntaxe

Afin de faire l'interface entre le code PHP et l'interface d'une page **OVIDENTIA**, un mini-langage de programmation existe dans le moteur de templates. Ainsi, il est possible dans un fichier template d'effectuer des tests de comparaisons ou d'égalité.

Exemple :

```
<!--#if BAB_SESS_LOGGED -->
Vous êtes authentifié
<!--#else BAB_SESS_LOGGED -->
Vous n'êtes pas authentifié
<!--#endif BAB_SESS_LOGGED -->
```

Le code précédent permet de tester si l'utilisateur courant est authentifié ou anonyme.

Voici les syntaxes courantes :

- Affichage d'une variable rendue disponible par PHP : { XXX }  
Exemple : dans le template topicsdisplay.html, la variable { artid } permet d'afficher l'identifiant d'un article
- Conditions et comparaisons : <!--#if XX "YY ZZZ" --> où XX est la valeur testée, YY est l'opérateur et ZZZ la valeur de comparaison  
Exemples :  
<!--#if tg "==" login" -->  
Vous êtes sur les pages d'authentification  
<!--#endif tg -->  
  
<!--#if nbsections ">= 1" -->  
Vous avez au moins 1 section affichée sur votre page  
<!--#endif nbsections -->

- Lancement d'une fonction : `<!--#in mafonction -->`  
Cette syntaxe permet d'exécuter une fonction définie en PHP, elle est généralement utilisée pour la génération de listes (tableaux, listes déroulantes...)  
Exemple :  
`<!--#in listevaleurs -->`  
    { valeur }  
`<!--#endin listevaleurs -->`

Pour plus de détails sur la syntaxe du moteur de templates, veuillez voir la documentation dédiée à la création des modules dans **OVIDENTIA** (AddonsTutorials).

## 5.2 - Les patrons d'affichage

Pour modifier l'apparence et la structure des éléments d'**OVIDENTIA**, il faut modifier le fichier template correspondant. Ainsi, la modification du fichier 'sectiontemplate.html' changera l'apparence de toutes les sections du portail. Si vous désirez proposer plusieurs apparences pour vos éléments, vous pouvez utiliser les patrons d'affichages.

Les patrons d'affichages existent pour les écrans suivants :

- affichage des sections personnalisées
- affichage des catégories d'articles
- affichage des thèmes d'articles (introductions des articles)
- affichage des articles (corps des articles)

### Créer une section



sélection du patron d'affichage dans une section personnalisée

Fonctionnement :

Pour créer des patrons d'affichage et ainsi présenter un choix d'interfaces à l'administrateur lors de la création de thèmes d'articles, de sections, ..., il est nécessaire d'ajouter ces patrons dans les fichiers templates correspondants. Voir la liste des fichiers templates modifiables en annexe pour connaître le fichier qui accueille les patrons que vous désirez ajouter.

Pour exemple, voici l'ajout d'un patron d'affichage pour les sections personnalisées :

- Le fichier template qui gère l'affichage des sections et qui accueille les patrons d'affichages est 'sectiontemplate.html'. Copiez ce fichier du noyau d'**OVIDENTIA** (racine d'Ovidentia/ noyau/skins/ovidentia/templates/) dans votre skin (racine d'Ovidentia/skins/votreskin/templates/).
- Vous trouverez dans le fichier original l'affichage nommé 'default' compris entre les balises begin et end :

```
<!--#begin default -->
[...]
```

```
<!--#end default -->
```

- Dupliquez l'affichage default en prenant soin de le renommer par votre patron d'affichage :

```
<!--#begin default -->
```

```
[...]
```

```
<!--#end default -->
```

```
<!--#begin monpatron -->
```

```
[...]
```

```
<!--#end monpatron -->
```

- Effectuez les modifications de structures ou de styles désirées dans le patron. Vous pouvez modifier la structure afin d'afficher systématiquement un logo, supprimer le titre des sections...
- Créer une nouvelle section en tant qu'administrateur sur le portail (lien Sections en section d'Administration) puis sélectionnez le nouveau patron

Remarque :

Pour les patrons d'affichage des thèmes d'articles, il est nécessaire de créer 2 affichages car le fichier template topicsdisplay.html gère 2 écrans (thème d'articles et corps d'un article). Ainsi le fichier contiendra ces codes :

```
<!--#begin head_default -->
```

```
<!--#end head_default -->
```

```
<!--#begin body_default -->
```

```
<!--#end body_default -->
```

```
<!--#begin head_monpatron -->
```

```
<!--#end head_monpatron -->
```

```
<!--#begin body_monpatron -->
```

```
<!--#end body_monpatron -->
```

## 6 - Le langage OVML en détails

### 6.1 - Syntaxe

Le langage OVML permet de dynamiser les pages du portail en permettant au webmaster un accès au contenu du portail. Ainsi le webmaster pourra accéder à la liste complète des informations publiées et les restituer afin de donner des points d'entrée à son site.

Exemples : affichage de la liste des catégories d'articles dans une section, affichage des derniers articles publiés en pages d'accueil...

Le langage a une syntaxe proche de l'HTML et du XML par ses balises. Le code doit être placé dans un fichier situé dans le répertoire ovml de votre skin (racine d'Ovidentia/skins/votreskin/ovml/).

Il existe plusieurs syntaxes d'appels selon le contexte :

- Afficher le contenu interprété d'un fichier OVML dans le corps d'**OVIDENTIA** :  
index.php?tg=oml&file=monfichier.ovml  
index.php?tg=oml&file=agendas/monfichier.ovml si le fichier est situé dans un sous-répertoire 'agendas' du répertoire ovml de votre skin
- Afficher le contenu interprété d'un fichier OVML en pleine page :  
index.php?tg=oml&file=monfichier.ovml&echo=1
- Afficher le contenu interprété d'un fichier OVML par l'éditeur wysiwyg (par exemple, dans une section ou dans un article) :  
\$OVML(monfichier.ovml)  
\$OVML(agendas/monfichier.ovml) si le fichier est situé dans un sous-répertoire 'agendas' du répertoire ovml de votre skin
- Afficher le contenu interprété d'un fichier OVML depuis un fichier template (par exemple dans page.html) :

```
{ $OVML(monfichier.ovml) }  
{ $OVML(agendas/monfichier.ovml) } si le fichier est situé dans un sous-répertoire 'agendas' du  
répertoire ovml de votre skin
```

Voici les syntaxes courantes :

- Création d'une variable : `<OFPutVar name="compteur" value="0">`  
Après la déclaration de la variable 'compteur', sa valeur peut s'afficher avec la syntaxe `<Ovcompteur>`
- Tests de comparaisons : `<OCIfEqual>`, `<OCIfNotEqual>`, `<OCIfGreaterThan>`...  
Exemples :  
`<OCIfEqual expr1="<OVBAB_SESS_USERID>" expr2="6">`  
Vous êtes l'utilisateur ayant l'identifiant 6  
`</OCIfEqual>`
- Affichage d'une variable globale : `<OFPutVar>` permet de définir la variable afin de récupérer sa valeur  
Exemple :  
`<OFPutVar name="babSkin">`  
``

Pour plus de détails sur la syntaxe et les possibilités du langage, veuillez voir les 2 documentations existantes (ovi-ovml et ovi-ovml utilisation).

## 6.2 - Pages d'accueil

Les pages d'accueil publique et privée d'**OVIDENTIA** peuvent être personnalisées par le webmaster grâce au langage OVML. Il existe 2 fonctionnements utilisables pour la gestion des pages d'accueil :

1) Si vous supprimez les fichiers `private.html` et `public.html` de votre skin (racine d'Ovidentia/skins/votreskin/ovml/), **OVIDENTIA** utilisera le template `'entry.html'` qui affichera les articles proposés en pages d'accueil (voir les options des thèmes d'articles pour plus de détails).

2) Si vous voulez personnaliser les pages d'accueil, vous pouvez utiliser les fichiers `'public.html'` (gestion de la page d'accueil publique) et `'private.html'` (gestion de la page d'accueil privée) présents dans votre skin (racine d'Ovidentia skins/votreskin/ovml/).

Les 2 fichiers `'private.html'` et `'public.html'` peuvent contenir des codes html classiques (html, javascript, css...) et ont aussi l'avantage d'y ajouter l'interprétation de codes OVML.

Exemple :

Ce code affichera les 5 derniers articles publiés (introduction et corps) en tenant compte des droits d'accès :

```
<OCRecentArticles last="5">  
  <p>  
    <h2><OVArticleTitle></h2>  
    <OVArticleHead>  
    <OVArticleBody>  
  </p>  
</OCRecentArticles>
```

## 7 - Annexes

### 7.1 - Les variables globales

Les variables globales sont des informations disponibles dans tous les fichiers templates et OVML.

- Utilisation dans un fichier template :

Les variables globales sont automatiquement disponibles dans un fichier template. Utilisez la syntaxe courante des variables : { XXX } où XXX est le nom de la variable globale.

Remarque :

Toutes les variables globales de type booléennes s'utilisent dans des tests de conditions ou comparaisons, les variables ne s'affichent pas avec la syntaxe { XXX }.

Exemple :

Bienvenue { BAB\_SESS\_USER } affichera le nom et le prénom de l'utilisateur courant

- Utilisation dans un fichier OVML :

Pour utiliser une variable globale dans un fichier OVML, il faut dans un premier temps que la variable désirée soit placée dans le contexte OVML. Pour cela, l'utilisation de la syntaxe <OFPutVar> est nécessaire.

Remarque :

Les variables globales de type booléennes ayant la valeur 'true' afficheront la valeur '1' en OVML. La valeur 'false' ne ramène généralement aucune valeur en OVML. Il est préférable dans des tests d'égalité/inégalité de tester sur la valeur '1'.

Exemple :

```
<OFPutVar name="babSkin">
```

```

```

--> La première ligne déclare la variable globale babSkin afin d'en récupérer sa valeur. La deuxième ligne utilise babSkin (nom du skin courant) pour afficher une image.

Attention : veuillez respecter chaque caractère en tenant compte de la casse (majuscule/minuscule).

**BAB\_SESS\_LOGGED** : si la variable est à 'true', l'utilisateur courant est un utilisateur authentifié sur le site. Sinon, l'utilisateur courant est non authentifié (anonyme).

**BAB\_SESS\_USER** : nom et prénom de l'utilisateur authentifié. Si vide, l'utilisateur courant est non authentifié (anonyme).

**BAB\_SESS\_NICKNAME** : identifiant de l'utilisateur authentifié. Si vide, l'utilisateur courant est non authentifié (anonyme).

**BAB\_SESS\_USERID** : identifiant (entier) dans la base de données de l'utilisateur courant. Si vide, l'utilisateur courant est non authentifié (anonyme).

**BAB\_SESS\_EMAIL** : adresse e-mail de l'utilisateur courant. Si vide, l'utilisateur courant est non authentifié (anonyme).

**babInstallPath** : chemin du noyau d'**OVIDENTIA** .

**babSkin** : nom du skin courant utilisé.

**babSkinPath** : chemin du skin courant utilisé.

**babCssPath** : chemin du fichier css choisi par l'utilisateur courant.

**babScriptPath** : chemin du dossier script dans le noyau **OVIDENTIA** .

**babLanguage** : abrégé de la langue d'interface utilisée par l'utilisateur courant (en, fr, nl, nl-be, ...).

**babStyle** : nom du fichier CSS courant.

**babSiteName** : nom du site courant.

**babUrl** : url d'accès au site (comme écrit dans config.php et index.php).

**babDB** : nom de la base de données MySQL.

**babCurrentDate** : variable uniquement disponible dans les fichiers OVML, elle permet de récupérer la date

courante. Contrairement aux autres variables globales, elle ne doit pas être déclarée par la fonction OFPutVar afin de l'utiliser. babCurrentDate renvoie une date au format timestamp (Voir la documentation OVML pour un formatage différent).

## 7.2 - Les fichiers templates modifiables

Pour des raisons logiques d'évolution du portail, il n'est pas possible de modifier toutes les interfaces d'**OVIDENTIA** (fichiers templates). Ainsi seuls quelques fichiers templates sont modifiables par le webmaster. La structure de ces fichiers ne change pas au fur et à mesure des versions d'**OVIDENTIA** ce qui garantit au webmaster que son skin restera compatible avec les futures versions d'**OVIDENTIA** .

Voici une liste des fichiers templates modifiables et leurs natures :

**adminsection.html** : section administrateur

**articlestemplate.html** : modèle d'articles pour les thèmes d'articles

**config.html** : balises méta

**forumssection.html** : section qui liste les forums

**mailtemplate.html** : mails de notification

**mailinfo.html** : personnalisation des informations de notification

**montha.html** : section calendrier

**page.html** : page principale lançant les sections et le corps d'**OVIDENTIA**

**sectiontemplate.html** : sections, patrons d'affichages des sections

**topcatsection.html** : section de catégories de thèmes

**topicsdisplay.html** : liste des articles (introduction + lien "lire la suite"), corps des articles, patrons d'affichage des thèmes d'articles

**topicssection.html** : section de catégories de thèmes si thèmes existants, patrons d'affichages de section pour les catégories

**usersection.html** : section utilisateur

**warning.html** : n'est plus utilisé dans **OVIDENTIA** (gère les 2 variables Titre et Message d'erreur)

**topcatdisplay.html** : liste des thèmes dans une catégorie, patrons d'affichage pour les catégories

**entry.html** : affichage des articles en page d'accueil lorsque les fichiers private.html ou public.html sont absents du skin

## 7.3 - Les variables utilisables dans mailinfo.html

Dans le fichier de mailinfo.html, pour personnaliser les objets des emails, ajouter pour chaque email un **template** xxxxx\_subject. Par exemple pour personnaliser l'objet de l'email du **forum**, ajouter:

On peut mettre le fichier mailinfo.html dans le répertoire de skin et le personnaliser. On n'est pas obligé de mettre tout le contenu du fichier mailinfo.html.

On ne met que la partie personnalisée.

### 7.3.1 - Pour personnaliser l'objet de l'email des notifications de nouveaux articles

```
<!--#begin notifyarticle_new_ subject -->  
ma personnalisation  
<!--#end notifyarticle_new_ subject -->
```

Les variables disponibles :

- babtpl\_topicname
- babtpl\_authorname
- babtpl\_articledatetime
- babtpl\_articledate
- babtpl\_articletime
- babtpl\_articletitle

- babtpl\_articleid
- babtpl\_articletopicid

### **7.3.2 - Pour personnaliser l'objet de l'email des notifications de mise à jour des articles**

```
<!--#begin notifyarticle_update_subject -->  
ma personnalisation  
<!--#end notifyarticle_update_subject -->
```

Les variables disponibles :

- babtpl\_topicname
- babtpl\_authorname
- babtpl\_articledatettime
- babtpl\_articledate
- babtpl\_articletime
- babtpl\_articletitle

### **7.3.3 - Pour personnaliser l'objet de l'email des notifications des approbateurs d'un article**

```
<!--#begin articlewait_subject -->  
ma personnalisation  
<!--#end articlewait_subject - ->
```

Les variables disponibles :

- babtpl\_topicname
- babtpl\_authorname
- babtpl\_articledatettime
- babtpl\_articledate
- babtpl\_articletime
- babtpl\_articletitle

### **7.3.4 - Pour personnaliser l'objet de l'email des notifications pour les commentaires**

```
<!--#begin commentwait_subject -->  
ma personnalisation  
<!--#end commentwait_subject - ->
```

Les variables disponibles :

- babtpl\_topicname
- babtpl\_authorname
- babtpl\_articledatettime
- babtpl\_articledate
- babtpl\_articletime
- babtpl\_articletitle
- babtpl\_commentssubject

### **7.3.5 - Pour personnaliser l'objet de l'email des notifications de d'acceptation/refus d'un article**

```
<!--#begin confirmarticle_ accepted_subject -->  
ma personnalisation  
<!--#end confirmarticle_ accepted_subject -->
```

ou

```
<!--#begin confirmarticle_ refused_subject -->  
ma personnalisation  
<!--#end confirmarticle_ refused_subject -->
```

Les variables disponibles :

- babtpl\_topicname
- babtpl\_authorname
- babtpl\_articledatetime
- babtpl\_articledate
- babtpl\_articletime
- babtpl\_articletitle

### **7.3.6 - Pour personnaliser l'objet de l'email des notifications des nouvelles contributions de forum**

```
<!--#begin newpost_subject -->  
ma personnalisation  
<!--#end newpost_subject -->
```

Les variables disponibles :

- babtpl\_forum (Le nom du forum)
- babtpl\_thread (Le titre du sujet)
- babtpl\_author (l'auteur)